

МОДИФИКАЦИЯ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ С ОГРАНИЧЕННЫМИ РЕСУРСАМИ

А.С. Васильев, В.Г. Матвейкин

Кафедра «Информационные процессы и управление», ГОУ ВПО «ТГТУ»

Ключевые слова и фразы: генетический метод; календарное планирование.

Аннотация: Рассматривается известная NP-трудная задача календарного планирования с ограниченными ресурсами. Предлагается модификация генетического алгоритма для решения этой задачи.

Исследования задач календарного планирования (ЗКП) проводятся уже более 40 лет. Актуальность этих исследований обусловлена важными практически приложениями. С математической точки зрения, исследуемые задачи относятся к числу NP-трудных задач дискретной оптимизации. Для их решения разрабатывались как точные методы, основанные на методе ветвей и границ, так и приближенные детерминированные и вероятностные методы. Задачи календарного планирования отражают процесс распределения во времени ограниченного числа ресурсов для выполнения проекта, состоящего из заданного множества взаимосвязанных работ. Помимо широкого практического применения, данная задача представляет интерес с научной точки зрения, поскольку календарное планирование содержит достаточно богатый класс известных сложных задач дискретной оптимизации. В качестве примера можно привести ряд задач теории расписаний, являющихся частными случаями рассматриваемой задачи.

В задаче календарного планирования с ограниченными ресурсами (ЗКПОР) задано множество работ, связанных друг с другом условиями предшествования (рис. 1). Эти условия задают технологию выполнения работ и на множестве работ порождают частичный порядок. Для каждой работы задана длительность ее выполнения и объемы потребляемых ресурсов. Суммарный объем каждого ресурса считается известным в каждый момент времени. Все ресурсы являются нескладируемыми, то есть неиспользованные ресурсы пропадают как, например, рабочее время. Выделение и потребление ресурсов предполагается равномерным. Требуется найти расписание (рис. 2) выполнения работ, удовлетворяющее условиям предшествования, ограничениям по ресурсам и минимизирующее время окончания всех работ.

Обозначим через $J = \{1, \dots, n\} \cup \{0, n+1\}$ множество работ, в котором 0-я и $(n+1)$ -я работы фиктивны и задают начало и завершение всего комплекса работ. Длительность j -й работы обозначим через $p_j, j \in J$. Считаем, что $p_j > 0, j \in J$ – целые неотрицательные числа и $p_0 = p_{n+1} = 0$. Отношения предшествования на множестве J зададим множеством пар $C = \{(i, j) \mid i \text{ предшественник } j\}$. Если $(i, j) \in C$, то работа j не может начаться раньше завершения работы i . В множестве C содержатся все пары $(0, j)$ и $(j, n+1), j = 1, \dots, n$. Через P_j обозначим множество

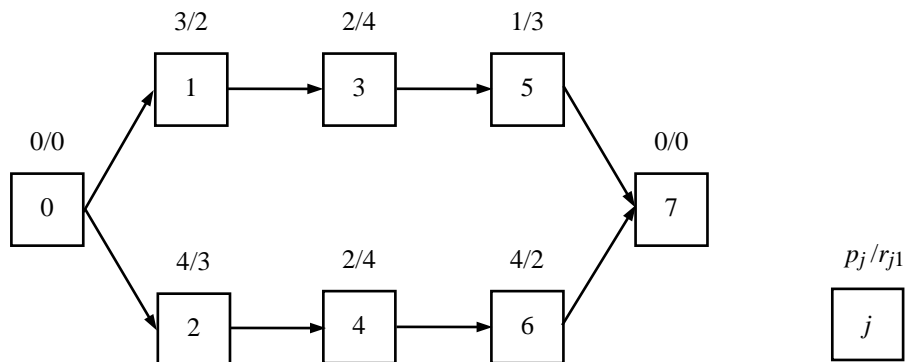


Рис. 1. Пример задачи из $n = 6$ работ, использующие $K = 1$ тип ресурса общим количеством $R_1 = 4$

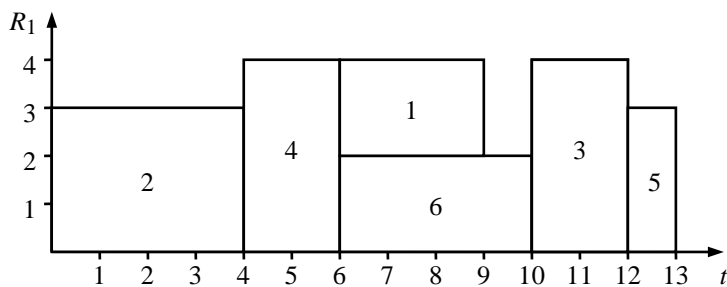


Рис. 2. Пример расписания (условия задачи см. на рис. 1)

непосредственных предшественников работы j . Через $K = \{1, \dots, w\}$ обозначим множество ресурсов, необходимых для выполнения рассматриваемых работ. Все ресурсы предполагаются возобновляемыми и нескладируемыми, то есть в каждый момент времени выделяется фиксированный объем ресурсов каждого типа, а остатки ресурсов пропадают. Будем считать, что объем выделяемого ресурса $R_k > 0$, $k \in K$, есть величина постоянная. Через $r_{jk} > 0$ обозначим объем k -го ресурса, необходимый для выполнения j -й работы.

Введем переменные задачи. Через $s_j > 0$ обозначим момент начала выполнения j -й работы. Предполагаем, что работы выполняются без прерывания, и момент окончания j -й работы определяется равенством $c_j = s_j + p_j$. Через $A(t) = \{j \in J \mid s_j \leq t < c_j\}$ обозначим множество работ, выполняемых в момент времени t . Время завершения проекта обозначим через $T(S)$. Эта величина соответствует моменту завершения последней работы проекта, то есть $T(S) = c_{n+1}$. При этих обозначениях задача календарного планирования с ограниченными ресурсами записывается следующим образом.

Найти $T(S) \rightarrow \min$ при условиях:

$$\begin{aligned}
 c_i &\leq s_j, i \in P_j, \\
 \sum_{j \in A(t)} r_{jk} &\leq R_k, k \in K, t \geq 0; \\
 s_j &\geq 0, j \in J.
 \end{aligned}
 \tag{1}$$

Целевая функция задает время завершения всего комплекса работ. Первое неравенство задает условия предшествования. Второе неравенство требует выполнения ресурсных ограничений.

Доказано, что сформулированная задача является NP-трудной, таким образом разработка эвристических методов [5] является, по-видимому, наиболее перспективным направлением для решения данной задачи.

В большинстве работ, в которых изучаются алгоритмы решения ЗКПОР, используется кодировка решения в виде списка $L = (j_1, \dots, j_n)$. Предполагается, что списки согласованы с условиями предшествования, то есть для любой пары $(i, j) \in C$ позиция работы j больше позиции работы i . По списку можно построить расписание. Для этого используются три подхода построения расписаний работ с использованием последовательного, параллельного и T-позднего декодеров.

Первый декодер, согласно списку, последовательно вычисляет наиболее раннее время начала выполнения каждой работы, учитывая ресурсные ограничения.

Второй декодер вычисляет наиболее раннее время, когда можно начать хотя бы одну из невыполненных работ и для этого момента времени принимает решение о начале выполнения очередных работ из списка. Произвол в выборе таких работ открывает широкие возможности для совершенствования декодеров этого типа.

T-поздний декодер действует аналогично первому, но использует список в обратном порядке. Фиксируется длина расписания T , и с учетом ресурсных ограничений для каждой работы вычисляется наиболее позднее время ее окончания. Как уже было сказано выше, условия предшествования учитываются при составлении списка и выполняются автоматически.

Последовательный декодер строит расписание, в котором ни одна работа не может начаться раньше установленного ей срока без нарушений условий предшествования или ограничений по ресурсам. Такие расписания называются активными. Известно, что среди активных расписаний есть и оптимальное расписание.

T-поздний декодер строит расписание, в котором ни одна работа не может закончиться позже установленного ей срока без нарушений либо условий предшествования, либо ограничений по ресурсам. Такие расписания называются T-поздними. Если ограничения по ресурсам не зависят от времени, то среди T-поздних расписаний найдется оптимальное.

Для решения задач теории расписаний интенсивно исследовались быстрые полиномиальные алгоритмы, основанные на приоритетных правилах. Согласно этому подходу среди множества работ, доступных к выполнению, выбиралась одна работа по заданному критерию, например, работа с минимальной длительностью, минимальным числом предшественников, или другому критерию. Понятно, что ни одно из таких правил не гарантирует получения оптимального решения, если задача является NP-трудной.

В настоящий момент разработано множество эвристических методов, основанных на применении генетических алгоритмов (ГА) и их модификаций [1–3], позволяющих находить близкое к оптимальному решение ЗКПОР.

Общая схема работы ГА применительно к ЗКПОР следующая.

1. Формируется начальная популяция, состоящая из некоего числа N допустимых согласно условий предшествования расписаний, представленных в виде списков работ в порядке их предпочтения.

2. Из популяции выбираются пары решений, и над ними выполняется операция скрещивания. В большинстве методов от пары решений получается пара новых решений, которая образует новое поколение, состоящее из N решений.

3. Для нового поколения решений применяется оператор мутации. Суть данного шага заключается в искусственном изменении порядка следования операций в списке. Для каждого решения I в списке работ L^I для позиций $i = 1, \dots, J-1$, операции j_i^I и j_{i+1}^I с вероятностью p_{mut} меняются местами, но только в том случае, если это не приводит к нарушению условий предшествования. Таким обра-

зом, получается новая популяция, состоящая из N особей прошлого поколения и N особей нового. Итого на данном шаге имеем популяцию, состоящую из $2N$ особей.

4. К особям популяции применяется оператор отбора. Для этого по каждому списку работ, входящему в состав популяции, одним из декодеров (например, при помощи активного декодера) строится расписание, по которому вычисляется качественный критерий полученного расписания, например, время завершения c_{n+1} . Список, состоящий из особей популяции, сортируется по убыванию критерия качества и удаляется N особей из хвоста этого списка. Таким образом, размер популяции приводится к исходному, состоящему из N лучших из $2N$ полученных на шаге 3 особей. Также рекомендуется предварительно очистить популяцию от клонов, то есть одинаковых особей.

5. Шаги 2–4 повторяются до тех пор, пока не выполнится критерий остановки работы алгоритма. Это может быть как достижение заданного количества поколений (повторов выполнения шагов 2–4), так и достижение процессорного времени, отведенного на решение задачи, или другие критерии.

В приведенной схеме видно, что на эффективность работы алгоритма влияют следующие факторы:

- 1) алгоритм формирования начальной популяции;
- 2) способ скрещивания особей: одноточечный, двухточечный или многоточечный;
- 3) размер популяции N ;
- 4) вероятность мутации p_{mut} ;
- 5) критерий, выбранный для оценки качества расписания;
- 6) декодер, при помощи которого строится расписание.

Как правило, выбор этих параметров осуществляется эмпирически на базе численных экспериментов, в ходе которых подбираются такие значения этих параметров, которые позволяют найти компромисс между точностью и быстродействием алгоритма для заданной размерности задачи и средней плотности загрузки ресурсов.

Так, например, в разных литературных источниках наблюдаются совершенно разные взгляды на выбор размера популяции, но, при этом, везде размер популяции остается неизменным в ходе эволюции. В результате наблюдений за работой ГА было замечено, что если в случайно сформированную начальную популяцию добавить заведомо известное хорошее решение – так называемого искусственного лидера, можно заметить, что в ходе эволюции в каждой новой популяции наиболее хорошие показатели критерия качества будут содержаться у особей, производных от этого лидера или его потомков. Да и в целом, получение нового абсолютного лидера в популяции наиболее вероятно от особей, имеющих наивысшее значение критерия качества, а соответственно производство потомства от особей с низкими показателями качества в популяции менее перспективно, так как вероятность получения хорошего решения в этом случае существенно ниже.

Предлагается менять размер популяции в зависимости от количества лидирующих особей в популяции в соответствии со следующей схемой, применяемой на шаге 4 общей схемы работы ГА.

1. Найти особь с наивысшим значением критерия качества – лидера.
2. Подсчитать количество X особей, имеющих значение критерия как у лидера.
3. Если $X/(2N) > 0,5$, то это значит, что в популяции в настоящий момент содержится более половины лидеров и рекомендуется не производить отсев, тем самым увеличив размер популяции в 2 раза. Если $X/(2N) < 0,1$, то это значит, что в популяции в настоящий момент содержится менее 10 % лидирующих особей и, чтобы не тратить лишнее процессорное время, рекомендуется отсеять $(3/2)N$, округленное до ближайшего целого, особей, тем самым, сократив размер популяции

в 2 раза. В остальных случаях, а также тогда, когда увеличение или уменьшение популяции приведет к нарушению ограничений на максимально и минимально возможный размер популяции – оставить размер популяции на прежнем уровне, удалив N особей.

Максимально возможный размер популяции устанавливается в зависимости от объемов доступной оперативной памяти, а также временных ограничений на выполнение 1 шага эволюции.

Минимально возможный размер популяции не рекомендуется делать менее, чем n – число операций, составляющих расписание.

Численные эксперименты, проводимые на тестовом наборе данных PSPLib [4], показали сокращение объемов вычислений для получения оптимального решения ЗКПОР на тестовых примерах с $n = 30$, а для задач с $n = 60$ и более также наблюдалась более высокая эффективность поиска по сравнению с ГА, использующим постоянный размер популяции.

Список литературы

1. Hartmann S.A. Competitive Genetic Algorithm for Resource-Constrained Project Scheduling, *Naval Research Logistics* 45:733–750, 1998.
2. Hartmann S.A Self-Adapting Genetic Algorithm for Project Scheduling under Resource Constraints, *Naval Research Logistics* 49:433–448, 2002.
3. Holland H.J. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
4. Kolisch R., Schwindt C., Sprecher A. Benchmark Instances for Project Scheduling Problems. *Project Scheduling. Recent Models, Algorithms and Applications*. Boston : Kluwer Acad. Publ. 197–212, 1999.
5. Kolisch R., Hartmann S. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In: Weglarz, J. (Ed.), *Project Scheduling – Recent Models, Algorithms and Applications*, Kluwer Academic Publishers, Boston, 1999. – P. 147–178.

Modification of Genetic Algorithm for the Solution to the Problem of Calendar Planning with Limited Resources

A.S. Vasilyev, V.G. Matveikin

Department "Information Processes and Control", TSTU

Key words and phrases: calendar planning; genetic method.

Abstract: The paper studies the well-known NP-difficult problem of calendar planning with limited resources. The modification of genetic algorithm for the solution to this task is offered.

Modifikation des genetischen Algorithmus für die Lösung der Aufgabe der Kalenderplanung mit den beschränkten Ressourcen

Zusammenfassung: Es wird die bekannte NP-schwere Aufgabe der Kalenderplanung mit den beschränkten Ressourcen betrachtet. Es wird die Modifikation des genetischen Algorithmus für die Lösung dieser Aufgabe vorgeschlagen.

Modification de l'algorithme génétique pour la solution du problème du calendrier aux ressources limitées

Résumé: Est examiné le fameux problème NP-difficile du calendrier aux ressources limitées. Est proposée la modification de l'algorithme génétique pour la solution de ce problème.