УДК 004.02: 004.942:007

М.В. САВЕЛЬЕВ*

О СЛОЖНОСТИ ОДНОЙ ЗАДАЧИ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

*Институт проблем математических машин и систем НАН Украины, Киев, Украина

Анотація. *У статті розглядається одне із завдань комбінаторної оптимізації, пов'язане з виникаючою на практиці проблемою розстановки персоналу по безлічі робіт у разі, якщо на безлічі робіт можуть існувати обмеження на порядок їх виконання, а персонал має нерівнозначну підготовку. Наводиться математичне формулювання такого завдання і показується його зведення до NP-повної задачі "про ранці» для випадку, коли на підмножині робіт відсутні обмеження слідування.*

Ключові слова: *комбінаторна оптимізація, складність, NP-задачі, теорія розкладів.*

Аннотация. *В статье рассматривается одна из задач комбинаторной оптимизации, связанная с возникающей на практике проблемой расстановки персонала по множеству работ в случае, если на множестве работ могут существовать ограничения на порядок их выполнения, а персонал имеет неравнозначную подготовку. Приводится математическая формулировка такой задачи и показывается ее сводимость к NP-полной задаче «о ранце» для случая, когда на подмножестве работ отсутствуют ограничения следования.*

Ключевые слова: *комбинаторная оптимизация, сложность, NP-задачи, теория расписаний.*

Abstract. *The article considers one of the tasks of combinatorial optimization, linked with the practice problems of arrangement of staff with different competence on a variety of jobs that have restrictions on their execution order and the staff has inadequate preparation. A mathematical formulation of this problem is provided. It is shown the reduction to NP-complete "knapsack" problem for the case when the following restrictions on the subset are absent.*

Keywords: *combinatorial optimization, complexity, NP-tasks, scheduling theory.*

1. Введение

При управлении проектами, когда группа людей осуществляет совместную деятельность, возникает задача оценки сроков выполнения проекта, то есть расчета времени, которое затратит эта группа людей для достижения проектных целей. Зачастую именно от сроков выполнения проекта зависит совокупный объем затрат, которые несут исполнители, заказчики и бенефицианты проекта.

Очевидно, что сроки будут зависеть от объема выполняемых работ, а также от оптимального распределения работ по исполнителям, учитывающего их индивидуальные способности к выполнению конкретного задания.

Данная проблема является особенно актуальной в современной экономике, эксплуатирующей не столько физические, сколько интеллектуальные качества людей, и связана с невозможностью оперативной замены исполнителя на равноценного или лучшего по качеству.

В общем виде проблема распределения работ по исполнителям является разновидностью проблемы выбора и/или расположения некоторого, обычно конечного, множества в соответствии с заданными правилами с целью минимизировать (максимизировать) некоторую целевую функцию. Согласно Лазареву А.А. [1], такая проблема относится к классу задач комбинаторной оптимизации и входит в раздел дискретной математики – теория расписаний.

Цель данной статьи – показать сложность одной из задач комбинаторной оптимизации, связанной с возникающей на практике задачей расстановки неравнозначного по качеству персонала по множеству работ, имеющих ограничения на порядок их выполнения.

2. Анализ последних достижений и публикаций

Проблема проектного управления и планирования ресурсов возникла в конце XIX – начале XX века в ходе формирования первых индустриальных обществ. Здесь следует отметить работы Гантта Г.Л. [2], Тейлора Ф.В. [3], в которых была сформулирована необходимость в научных подходах и приведены практические методы планирования. С середины XX века начались активные теоретические исследования задач планирования и распределения ресурсов, получившие в 1956г. наименование «теория расписаний» Белманна Р. [4]. Здесь следует отметить работы [5–7]. Важнейшими вопросами в исследовании таких задач стали поиск эффективных алгоритмов решения, а в дальнейшем – установление сложности подобных таких задач. Важнейшей работой в этом направлении являются работы [8–10].

Следует отметить, что и в настоящее время продолжаются исследования отдельных проблем с целью определения их сложности [11–13].

Основным выводом данных исследований является то, что большинство проблем комбинаторной оптимизации относятся к классу NP-трудных. Тем не менее, для отдельных проблем были найдены эффективные алгоритмы с полиномиальной сложностью, например, задача о назначениях [1].

3. Формулировка проблемы и ее математическое описание

Определим $A = \{a_1, ...a_i..., a_n\}$ как множество работ, требуемых для завершения фазы жизненного цикла разработки продукта. Между некоторыми парами работ существует ограничение следования $a_i \to a_j$. Это означает, что работу a_j невозможно начать ранее работы a_i. Такое ограничение можно задать с помощью ориентированного графа $G = (V, D)$, где каждой вершине из множества $V = \{1, ..., n\}$ будет соответствовать одна работа из множества А, а множество дуг $D = \{(i, j) \mid i, j \in V;\ i \to j\}$ соответствует ограничению следования. Следует отметить, что такой граф G должен быть ацикличен.

Пусть $C = \{c_1, ...c_k..., c_s\}$ – множество сотрудников компании, где c_k – конкретный сотрудник, обладающий способностью выполнить работу a_i за время $t_{ik} > 0$. Формально, $t_{ik} \in (0, +\infty)$, поскольку сотрудник может не обладать необходимой квалификацией. В единичный момент времени один сотрудник может выполнять одну работу, прерывание работ не допускается. Очевидно, что такие длительности выполнения работ вышеназванными сотрудниками образуют множество T – суть декартово произведение множеств A и $C - T = A \times C = \{(a, c) \mid a \in A, c \in C\}$.

Тогда стоит задача оптимальной расстановки имеющихся сотрудников по работам так, чтобы общее время выполнения всех работ из множества *A* было минимальным и не нарушалось отношение следования.

4. Анализ задачи

Рассмотрим два крайних случая:

1. Существующие ограничения следования предписывают выполнять все работы последовательно от a_1 до a_n.

2. Отсутствуют ограничения следования, то есть все работы $a_1..a_n$ могут выполняться параллельно. Тогда решение задачи становится не столь очевидным.

Нетрудно показать, что все остальные варианты сводятся до этих двух крайних случаев при разбиении потока задач на части, где некий набор работ $a_i..a_j$ последователен или параллелен.

Вариант 1. Рассмотрим вариант задачи для случая существования ограничения на последовательное выполнение работ.

Принимая во внимание факт, что одна работа может выполняться только одним сотрудником, введем функцию $f(i,k)=1$ для случая, если работа a_i выполняется сотрудником c_k, и $f(i,k)=0$ во всех противных случаях. Тогда решение задачи о расстановке персонала сводится к поиску такой функции $f(i,k)$, когда значение выражения (1) будет минимальным при условии, что $\forall i \in [1,n] \exists k \in [1,s] | f(i,k)=1$.

$$T_{\min} = \sum_{k=1}^{s}\sum_{i=1}^{n} t_{ik} f(i,k) \to \min . \tag{1}$$

Для того, чтобы найти такую функцию, приведем следующие рассуждения. Разложим выражение (1) на 2 слагаемых:

$$\sum_{k=1}^{s}\sum_{i=1}^{n-1} t_{ik} f(i,k) + \sum_{k=1}^{s} t_{nk} f(n,k) . \tag{2}$$

Очевидно, что сумма будет минимальной, если оба слагаемых минимальны. При этом минимум второго слагаемого достигается в том случае, если найдено такое k, при котором значение t_{nk} минимально и $f(n,k)=1$. Иными словами, найден сотрудник с индексом k, который выполнит последнюю работу a_n за наименьшее время.

Продолжив подобное рассуждение для первого слагаемого, получаем, что для поиска всех значений k, где функция $f(i,k)=1$, необходимо найти $\min(t_{ik})$ для каждой работы a_i. Что в общем виде имеет линейную трудоемкость.

Вариант 2. Рассмотрим вариант задачи для случая отсутствия ограничения следования, дающего возможность выполнять работы параллельно.

Тогда задача сводится к поиску такой функции $f(i,k)$, когда значение выражения будет минимальным при тех же условиях, что $\forall i \in [1,n] \exists k \in [1,s] | f(i,k)=1$.

$$T_{\min} = \max_{1 \le k \le s}\left(\sum_{i=1}^{n} t_{ik} f(i,k)\right) \to \min . \tag{3}$$

Графически это можно проиллюстрировать следующим образом. Представим множество T в виде матрицы, где строки будут сотрудники, а столбцы – работы. Тогда значениями ячейки t_{ik} будет время, затрачиваемое сотрудником k на выполнение работы i. Отметим ячейки в таблице, где функция $f(i,k)=1$. Тогда сумма таких отмеченных ячеек будет минимальной длительностью выполнения всех работ для строго последовательного ограничения следования, а максимум из сумм по строкам – минимальной длительностью выполнения всех работ при полном отсутствии ограничения следования.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
c_1		+		+	+			
c_2	+						+	+
c_3			+			+		

Выполним поиск функции $f(i,k)$ для второго случая.

Если количество работ 1, то очевидно, что решение функции $f(1,k)=1$ для k, где t_{1k} будет минимально.

Если количество работ 2, тогда возможно два решения:

1. Функция $f(1,k)=1$ для k, где t_{1k} будет минимально, и $f(2,k')=1$ для k', где $t_{2k'}$ будет минимально. Общее время будет $\max(t_{1k}, t_{2k'})$.

2. Функция $f(1,k)=f(2,k)=1$ для некоторого k, если $\forall k' \neq k \mid (t_{1k}+t_{2k}) < t_{2k'}$. Общее время будет $t_{1k}+t_{2k}$.

Допустим, что существует оптимальная функция $f(i,k)$ для $n-1$ работ, такая что

$$T_{\min_{n-1}} = \max_{1 \le k \le s}(\sum_{i=1}^{n-1} t_{ik} f(i,k)) \to \min . \tag{4}$$

Рассмотрим случай с добавлением работы n. Тогда возможен следующий вариант решения: $\exists k$ такое, что $\forall i < n(\sum_{i=1}^{n-1} t_{ik} f(i,k) + t_{nk}) \to \min$. Однако данное решение хоть и является допустимым, но не является оптимальным. Покажем это.

Рассмотрим два таких решения $T_{\min_{n-1}} < T'_{\min_{n-1}}$, характеризующимися функциями $f(i,k)$ и $f'(i,k)$, такими, что для $k \neq k' \exists i > 0 f'(i,k) = 1 => f'(i,k') = 0$ и $f(i,k) = 0$, но $f(i,k') = 1$, так что

$$T_{\min_{n-1}} = \sum_{i=1}^{n-1} t_{ik} f(i,k') < \sum_{i=1}^{n-1} t_{ik} f'(i,k') = T'_{\min_{n-1}},$$

тогда $$(\sum_{i=1}^{n-1} t_{ik'} f(i,k') - \sum_{i=1}^{n-1} t_{ik'} f'(i,k')) = t_{ik'},$$

а $$(\sum_{i=1}^{n-1} t_{ik} f'(i,k) - \sum_{i=1}^{n-1} t_{ik} f(i,k)) = t_{ik}.$$

Охарактеризуем работу a_n следующим кортежем длительностей ее выполнения сотрудниками из множества *C*, где длительность работы сотрудником $k' = (t_{ik'} + t_{ik})$, а остальные – бесконечно большие числа. Тогда

$$T_{\min_n} = \sum_{i=1}^{n} t_{ik} f(i,k') > \sum_{i=1}^{n} t_{ik} f'(i,k') = T'_{\min_n}. \tag{5}$$

Тем самым показано, что можно подобрать такой пример, когда в случае добавления новой работы при отсутствии ограничения следования, которое дает возможность выполнять работы параллельно, нельзя воспользоваться оптимальным решением, полученным на предыдущем шаге.

В случае отсутствия ограничения следования, дающего возможность выполнять работы параллельно, не удается «сходу» построить быстрый алгоритм решения, тогда имеет смысл определить, относится ли данная задача к классу *NP*. В этом случае, в силу предположения, что $P \neq NP$ [14], найти точное ее решение за полиномиальное время не представляется возможным и потребуются приближенные алгоритмы.

5. Доказательство NP-полноты

Рассмотрим задачу из двух исполнителей, оба из которых обладают равными между собой компетенциями. Минимальное время выполнения всех работ, при возможности действовать двум исполнителям параллельно, будет достигнуто тогда, когда работы между ними можно разделить поровну, то есть $\forall i f(i,1) \neq f(i,2)$ и $\sum_{i=1}^{n} t_{i1} f(i,1) - \sum_{i=1}^{n} t_{i2} f(i,2) \to 0$, но в силу того, что $t_{i1} = t_{i2}$, это означает, что задача сводится к классической «задаче о ранце» (англ. – knapsack problem), относящейся к классу NP-полных задач [15].

6. Выводы и предложения

Несмотря на то, что большинство задач комбинаторной оптимизации относятся к классу NP, тем не менее, для каждого частного случая всегда следует делать анализ таких задач, поскольку в них можно выделить части или условия, когда решение может иметь более простую трудоемкость. Тогда полученный результат можно использовать для оптимизации алгоритмов и программного обеспечения, направленного на решение этих проблем.

Касательно рассмотренной в статье задачи, то для ее практического решения можно эффективно применять методы имитационного моделирования [16–17].

СПИСОК ЛИТЕРАТУРЫ

1. Лазарев А.А. Теория расписаний. Задачи и алгоритмы / А.А. Лазарев, Е.Р. Гафаров. – М.: Московский государственный университет им. М.В. Ломоносова, 2011. – 222 с.
2. Gantt H.L. A graphical daily balance in manufacture / H.L. Gantt // ASME Transactions. – 1903. – Vol. 24. – P. 1322 – 1336.
3. Taylor F.W. Shop Management / F.W. Taylor // ASME Transactions. – 1903. – Vol. 24. – P. 1337 – 1480.
4. Bellman R. Mathematical aspects of scheduling theory / R. Bellman // Journal of the Society of Industrial and Applaid Mathematics. – 1956. – Vol. 4. – P. 168 – 205.
5. Jackson J.R. Scheduling a production line to minimize maximum tardiness / J.R. Jackson // Research Report N 43 Management Science Research Project. University of California – 1955.
6. Задачи календарного планирования и методы их решения / В.В. Шкурба, Т.П. Подчасова, А.Н. Пшичук [и др.]. – Киев: Наукова думка, 1966. – 154 с.
7. Танаев В.С. Введение в теорию расписаний / В.С. Танаев, В.В. Шкурба. – М.: Наука, 1975. – 256 с.
8. Graham R.L. Bounds for certain multiprocessing anomalies / R.L. Graham // SIAM J. Appl. Math. – 1966. – Vol. 17. – P. 263 – 269.
9. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон; пер. с англ. М.: Мир, 1982. – 416 с.
10. Brucker P. Complexity of machine scheduling problems / P. Brucker, J.K. Lenstra, A.N.G. Rinnoy Kan // Math. Cent. Afd. Math Beslisk. – Amsterdam, 1975. – BW 43. – 29 p.
11. Posypkin M.A. Speedup estimates for some variants of the parallel implementations of the branch-and-bound method / M.A. Posypkin, I. Kh. Sigal // Computational Mathematics and Mathematical Physics. – 2006. – Vol. 46, N 12. – P. 2189 – 2202.
12. Gafarov E.R. On Lower and Upper Bounds for the Resource-Constrained Project Scheduling Problem / Gafarov E.R., Lazarev A.A., Werner F. – Magdeburg, 2010. – 27 p. – (Preprint 8/10, FMA, Otto-von-Guericke-Universitat).
13. Гафаров Е.Р. Доказательство NP-трудности частного случая задачи минимизация суммарного запаздывания для одного прибора / Е.Р. Гафаров, А.А. Лазарев // Известия АН: Теория и системы управления. – 2006. – № 3. – С. 120 – 128.
14. Cook S. The complexity of theorem proving procedures / S. Cook // Proc. of the Third Annual ACM Symposium on Theory of Computing. – 1971. – P. 151 – 158.
15. Krap R.M. Reducibility among combinatorial problems. University of California at Berkeley [Електронний ресурс] / R.M. Krap. – Режим доступу: http://www.cs.berkeley.edu/~luca/cs172/karp.pdf.
16. Литвинов В.В. The simulation model of ИТ-product (service) development by a «start-up» company growing inside an academic institution / В.В. Литвинов, М.В. Савельев // Математичні машини і системи. – 2015. – № 4. – С. 92 – 99.
17. Савельєв М.В. Імітаційна модель процесу розробки IT-продукта силами університетської команди / М.В. Савельєв, А.В. Дерлеменко // Інформаційні технології: теорія, інновації, практика: матеріали конф. – Полтава: Полт. НТУ ім. Ю. Кондратюка, 2015. – С. 105 – 107.

Стаття надійшла до редакції 18.10.2016