Планирование проекта с учётом навыков сотрудников в условиях нечёткой информации

В. В. Коротков, email: chasecrunk@gmail.com

Воронежский государственный университет

Аннотация. *В данной работе рассматривается задача построения календарного плана проекта и распределения неоднородных человеческих ресурсов, обладающих различными наборами навыков. Приводится формальная модель задачи, учитывающая неопределённости в оценках продолжительностей работ и уровней владения навыками при помощи аппарата нечётких чисел. Описывается генетический алгоритм решения задачи.*

Ключевые слова: *календарное планирование, планирование проекта, комбинаторная оптимизация, нечёткие числа, генетический алгоритм.*

Введение

Методы календарно-сетевого планирования используются для построения оптимального по времени графика и оценки сроков выполнения работ проекта. Если при этом учитываются имеющиеся ресурсные ограничения, то такую задачу называют RCPSP (Resource Constrained Project Scheduling Problem).

Классическая формулировка данной задачи предполагает однородность и взаимозаменяемость ресурсов одного типа. Это не позволяет учитывать квалификацию сотрудников, а также формировать их рабочие графики. Поэтому в некоторых областях, таких как управление ИТ-проектами, получила распространение модификация, учитывающая специализации сотрудников (Multi-Skill RCPSP) [1-2]. Однако эта модифицированная модель задачи предполагает наличие нескольких дискретных уровней мастерства, обычно соответствующих некой общепринятой классификации. Например, в сфере разработки принято выделять уровни: junior, middle, senior. Такой подход является слишком грубым и не позволяет учесть индивидуальные наклонности отдельных сотрудников.

Ещё одной проблемой, возникающей в процессе управления любым проектом, является неопределённость, вызванная недостаточностью информации и влиянием множества случайных факторов. Эта проблема особенно актуальна на ранних этапах

планирования проекта, когда при этом необходимо принимать критически важные управленческие решения. Для учёта этих неопределённостей обычно используют вероятностные или нечёткие модели [3-4].

1. Постановка задачи

Сформулируем нечёткую постановку задачи, которая бы учитывала неопределённость в оценках основных параметров и индивидуальные наклонности сотрудников. Преимущество нечёткого подхода перед вероятностным в данном случае заключается в простоте получения оценок на основе экспертного анализа, опросов или исторической информации. Это особенно актуально при попытке оценить такую неоднозначную и субъективную величину, как уровень владения навыком. Таким образом, предлагается использовать аппарат нечётких чисел для представления уровней навыков и соответствующих требований. Опишем формальную постановку подобной задачи комбинаторной оптимизации.

Пусть J – множество работ проекта, $\tilde{\tau}_j$ – нечёткая оценка продолжительности работы $j \in J$, S – множество всех возможных навыков, L – множество сотрудников. Каждый сотрудник $l \in L$ владеет навыком $s \in S$ на уровне, который определяется нечёткой оценкой $\tilde{m}_{ls}$. Для выполнения каждой работы $j \in J$ необходимо r_{js} сотрудников, владеющих навыком $s \in S$ не хуже, чем на уровне $\tilde{d}_{js}$. Обозначим P_j – множество работ, предшествующих работе $j \in J$, а T – множество всех моментов времени в рамках горизонта планирования. Предполагается, что каждый сотрудник может быть единовременно занят в выполнении только одной работы и только относительно одного из своих навыков. Прерывания работ не допускаются.

Тогда для каждой работы $j \in J$ необходимо найти такое нечёткое время начала $\tilde{x}_j$ и множества выделенных на неё сотрудников $L_{js} \subset L, s \in S$, чтобы:

$$\max_{j \in J} \left(\tilde{x}_j + \tilde{\tau}_j \right) \to \min \tag{1}$$

при условии выполнения следующих ограничений:

$$\tilde{x}_j + \tilde{\tau}_j \le \tilde{x}_{j'} \qquad \forall j' \in J,\ j \in P_{j'} \tag{2}$$

$$\tilde{x}_j \ge \tilde{0} \qquad \forall j \in J \tag{3}$$

$$\tilde{m}_{lj} \geq \tilde{d}_{js} \qquad \forall j \in J, s \in S, l \in L_{js} \tag{4}$$

$$\left| L_{js} \right| = r_{js} \qquad \forall j \in J, s \in S \tag{5}$$

$$L_{js} \cap L_{js'} = \varnothing \qquad \forall j \in J; s, s' \in S, s \neq s' \tag{6}$$

$$\bigcup_{s \in S} L_{js} \cap \bigcup_{s \in S} L_{j's} = \varnothing \qquad \forall \tilde{t} \in T; j, j' \in A(\tilde{t}), j \neq j' \tag{7}$$

Здесь $\tilde{0}$ – нечёткий нуль, $A(\tilde{t}) = \left\{ j \in J \mid \tilde{x}_j \leq \tilde{t} \leq \tilde{x}_j + \tilde{\tau}_j \right\}$ – множество работ проекта, находящихся в процессе выполнения в нечёткий момент времени $\tilde{t}$.

2. Моделирование нечётких параметров

Будем полагать, что нечёткие параметры представлены в виде нечётких треугольных чисел. Тогда для некоторого параметра $\tilde{M}$ запись $\tilde{M} = (l, m, r)$ означает, что его функция принадлежности определяется как:

$$\mu_{\tilde{M}}(x) = \begin{cases} 0, & x < l, \\ \dfrac{x - l}{m - l}, & l \leq x \leq m, \\ \dfrac{r - x}{r - m}, & m \leq x \leq r, \\ 0, & x > r. \end{cases} \tag{8}$$

Для двух треугольных нечётких чисел $\tilde{M}_1 = (l_1, m_1, r_1)$ и $\tilde{M}_2 = (l_2, m_2, r_2)$ в соответствие с принципом обобщения Заде возможна следующая операция сложения:

$$\tilde{M}_1 + \tilde{M}_2 = (l_1 + l_2, m_1 + m_2, r_1 + r_2) \tag{9}$$

Для реализации алгоритма решения необходимо также задать операцию сравнения нечётких чисел. Существуют различные подходы к решению этой проблемы. Одними из наиболее распространённых являются интегральные методы, заключающиеся в сравнении нечётких чисел на основании неких средних чётких значений. Мы будем использовать метод предложенный в [3], при котором эти значения вычисляются следующим образом:

$$I_T\left(\tilde{M},\beta\right)=\frac{\beta}{2}(l+m)+\frac{1-\beta}{2}(m+r) \quad (10)$$

Параметр β здесь позволяет регулировать оптимистичность данной операции, что даёт большую гибкость для подстройки под конкретный случай. В общем же случае можно положить $\beta = 0.5$.

3. Алгоритм решения

Доказано, что задача RCPSP даже в своей классической постановке относится к классу NP-трудных [5]. Нахождение точного решения за разумное время возможно только для довольно малых экземпляров задач. Поэтому, как правило, применяются различные эвристические подходы, позволяющие быстро найти решение, достаточно близкое к оптимальному. Далее приводится описание компонентов генетического алгоритма, который использовался для решения задачи.

Решение кодируется хромосомой, которой соответствует ровно одно возможное расписание выполнения работ и распределение человеческих ресурсов. Каждый ген хромосомы содержит идентификатор работы и множества выделенных на неё сотрудников. При этом порядок генов в хромосоме отражает приоритет соответствующих работ и не нарушает ограничения предшествования. Расписание может быть получено из данного закодированного представления при помощи процедуры, называемой последовательная схема формирования расписания (Serial Scheduling Scheme) [6]. Начав с пустого расписания, новое промежуточное получают путем размещения следующей по приоритету работы на самое раннее время так, чтобы не нарушить ограничения предшествования и ресурсные ограничения. При этом для сравнения нечётких чисел и выполнения операций над ними используются подходы, описанные в пункте 2.

В качестве *функции приспособленности* используется целевая функция (1).

Для обеспечения разнообразия особей *начальная популяция* генерируется случайным образом. Для каждой хромосомы формируется случайный порядок работ путём их последовательного рандомизированного добавления из множества тех, предшественницы которых уже содержатся в решении. Для каждой работы $j \in J$ случайным образом берутся непересекающиеся подмножества ресурсов L_{js} мощностями r_{js} из множеств сотрудников, обладающих необходимыми уровнями навыков.

В работе был использован турнирный метод *селекции* с размером подгруппы равным 3.

Оператор скрещивания делит родительские хромосомы на две части в некоторой случайной точке. Все гены из первой части первого (второго) родителя в том же порядке перемещаются в первого (второго) наследника, а гены, соответствующие оставшимся работам, добавляются в том порядке, в каком они расположены во втором (первом) родителе.

Оператор мутации с некоторой заданной вероятностью срабатывания для каждого гена обменяет его с предыдущим местами, если это не нарушит ограничений предшествования. Аналогично с некоторой вероятностью может быть произведена замена множества выделенных на работу ресурсов на новое, полученное случайным образом с учётом ограничений.

4. Численный пример

Проиллюстрируем решение задачи на примере. В табл. 1 приведён список работ проекта с указанием необходимого числа исполнителей и пороговых значений навыков, требуемых для их выполнения. В табл. 2 – перечень сотрудников и оценок их навыков.

Таблица 1

Список работ проекта

№	Предшеств.	Продолжит.	Навыки, кол-во/требование		
			s_1	s_2	s_3
1		(12,13,14)	1 (1,2,3)	0	0
2		(15,16,17)	0	2 (2,4,5)	0
3		(18,19,25)	0	0	1 (1,1,3)
4	1,3	(1,2,5)	1 (1,2,2)	1 (1,1,2)	1 (1,2,3)
5	2	(16,17,19)	0	2 (1,2,3)	1 (2,2,3)
6	2,3	(2,3,5)	2 (1,2,4)	1 (2,2,3)	0
7	5,6	(4,6,7)	0	1 (3,4,4)	2 (1,2,3)

Таблица 2

Список сотрудников

№	Навыки		
	s_1	s_2	s_3
1	(1,2,3)	-	-
2	-	(3,5,5)	(1,2,3)
3	(1,2,4)	-	(2,3,3)
4	(2,2,3)	(4,4,5)	-
5	-	(2,2,3)	(2,2,3)

Для решения был использован описанный в пункте 3 генетический алгоритм. Были заданы следующие параметры алгоритма: размер популяции 100, число поколений 40, вероятность мутации 0.04.

Результаты решения задачи приведены в табл. 3. Полученные значения могут быть использованы для составления нечеткой диаграммы Ганта (см. рисунок), дающей наглядное представление о сетевом графике с учетом нечеткой неопределенности. Каждый четырехугольник на диаграмме определяет расположение во времени соответствующей работы одного из проектов. Треугольниками обозначены нечеткие времена начала и конца работ.

Таблица 3

Результаты решения задачи

№	Время начала	Исполнители
1	(0,0,0)	1
2	(0,0,0)	2
3	(12,13,14)	3
4	(32, 35, 44)	1,2,3
5	(15, 16, 17)	2,3
6	(30, 32, 39)	1,2
7	(33, 37, 49)	2,3

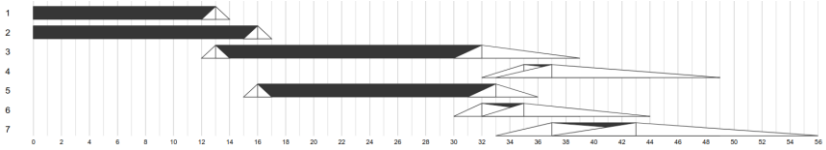

Рисунок. Нечёткая диаграмма Ганта полученного расписания

Заключение

Была рассмотрена модификация задачи RCPSP, учитывающая разнородность навыков сотрудников и нечёткость различных параметров модели. Приведён алгоритм эвристического поиска решения на базе генетического алгоритма.

Приведённый подход может использоваться на ранних этапах планирования проекта или когда существует значительная неопределённость в оценках основных параметров. Кроме того, модель позволяет определить приблизительные рабочие графики сотрудников, учитывающие их индивидуальные наклонности. Рассчитываемые сроки выполнения проекта и его работ представлены нечёткими треугольными числами, что позволяет помимо прочего количественно оценить степень неопределённости этих сроков, оценить риски и принять соответствующие управленческие решения.

Список литературы

1. Myszkowski, P. B. Co-evolutionary algorithm solving multi-skill resource-constrained project scheduling problem / P. B. Myszkowski, M. Laszczyk, D. Kalinowski // Federated Conference on Computer Science and Information Systems (Prague, 3-6 Sept. 2017). – Prague, 2017. – P. 75-82.

2. Skowronski, M. E. Tabu search approach for Multi-Skill Resource-Constrained Project Scheduling Problem / Federated Conference on Computer Science and Information Systems (Krakow, 8-11 Sept. 2013). – Krakow, 2013. – P. 153-158.

3. Hapke, M. Fuzzy priority heuristics for project scheduling / M. Hapke, R. Slowinski // Fuzzy Sets and Systems. – 1996. – Vol. 83. – Issue 3. – P. 291-299.

4. Wang, J. A fuzzy robust scheduling approach for product development projects / J. Wang // European Journal of Operational Research. – 2004. – Vol. 152. – Issue 1. – P. 180-194.

5. Blazewicz, J. Scheduling subject to resource constraints: classification and complexity / J. Blazewicz, J. K. Lenstra, A. H. G. Rinnooy Kan // Discrete Applied Mathematics. – 1983. – Vol. 5. – Issue 1. – P. 11-24.

6. Kolisch, R. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis / R. Kolisch, S. Hartmann // Project scheduling: Recent models, algorithms and applications. – Boston, 1998. – P. 147–178.