УДК 004.78; 681.5

М.В. ПОТАПОВА

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ ПРИ УПРАВЛЕНИИ ПОРТФЕЛЕМ ЗАКАЗОВ IT-КОМПАНИИ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

Рассмотрены факторы, влияющие на эффективность решений, связанных с календарным планированием программных проектов (КППП) в IT-компании. Показана недостаточная эффективность существующего подхода к компьютеризации КППП на основе использования стандартных офисных приложений. Разработана постановка задачи синтеза специализированной информационной технологии КППП, которая на формальном уровне сведена к задаче удовлетворения ограничений, в форме гиперграфа. Описаны стандартные и модифицированные процедуры анализа гиперграфа и представлена информационная технология КППП в виде упорядоченного набора процедур анализа совместности элементов гиперграфа ограничений.

Ключевые слова: *информационная технология, IT-компания, гиперграф ограничений, неопределенность, задача удовлетворения ограничений, программный проект.*

Введение

Эффективность функционирования компаний по производству программных продуктов существенным образом зависит от качества процессов планирования работ, связанных с реализацией портфеля заказов на выполнение программных проектов. В процессе планирования проекта определяются этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта [1]. Процесс планирования начинается с определения проектных ограничений. Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределение функций среди будущих исполнителей. Затем определяются этапы разработки и то, какие результаты должны быть получены по окончанию этих этапов. Следующим этапом является обеспечение цикличности планирования. При этом вначале разрабатывается график работ по выполнению проекта, затем, при возникновении расхождений между реальным и плановым ходом работ предусматривается ревизия первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены проектные ограничения.

План проекта должен четко показывать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В процессе составления графика весь массив работ, необходимых для реализации проекта, разбивается на отдельные этапы и оценивается время, требующееся для выполнения каждого этапа. Обычно многие этапы выполняются параллельно. График работ должен предусматривать это и распределять ресурсы между ними оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа – частая причина задержки выполнения всего проекта.

Разнообразие указанных выше факторов, непосредственным образом определяющих качество решений, принимаемых менеджерами IT-компаний при календарном планировании программных проектов (КППП), привело к широкому распространению целого ряда офисных приложений в качестве средств поддержки принятия решений по КППП [2]. При этом, применение этих средств в процессах КППП носит, как правило, фрагментарный характер, что не дает возможности реализовать КППП в рамках комплексного делопроизводства и документооборота в IT-компании.

Таким образом, единственной возможностью обеспечения должного уровня эффективных решений по КППП является разработка специализированной информационной технологии, охватывающей все этапы данного процесса с учетом релевантных факторов.

Цель статьи состоит в описании информационной технологии календарного планирования при управлении портфелем заказов IT-компании в условиях неопределенности.

1. Постановка задачи информационной технологии календарного планирования программных проектов в IT-компании

На содержательном уровне, задача информатизации процесса КППП может быть сформулирована следующим образом:

– исходные данные: поток программных проектов, находящихся в процессе разработки IT-компании; характеристики производственного персонала IT-компании; перечень факторов, непосредственно влияющих на выполнение каждого проекта;

– результат решения задачи: последовательность процедур в форме алгоритмов, реализующих весь комплекс бизнес-процессов, характерных для КППП;

– метод решения задачи: недоопределенные вычисления, программирование в ограничениях.

Формально, приведенная выше постановка задачи комплексной компьютеризации КППП может быть сведена к з**адаче удовлетворения ограничений** (ЗУО) [3]

$$M = (Z, D, C),$$

где Z – конечное множество переменных $\{x_1, x_2, \ldots, x_n\}$;

D – функция, которая отображает каждую переменную из множества Z на множество объектов произвольного типа:

D: Z → конечное множество объектов (заданного типа);

C – конечное (возможно пустое) множество ограничений на произвольном подмножестве переменных из Z.

Определение 1. Гиперграфом ограничений (ГО) ЗУО (Z, D, C) называется гиперграф, в котором каждая вершина соответствует переменной из Z, а каждое (гипер-)ребро соответствует ограничению из C.

Определение 2. ЗУО является **совместной в вершинах** тогда и только тогда, когда для каждой переменной все значения в области определения этой переменной удовлетворяют всем ограничениям над данной переменной:

$$\forall\, csp((Z,D,C)):$$
$$node\text{–}consistent((Z,D,C)) \equiv$$
$$\equiv (\forall x \in Z : (\forall v \in D_x : satisfies(< x, v >, C_x))).$$

Определение 3. Путь (x_0, x_1, …, x_m) в графе ограничений для ЗУО называется **совместным по пути** тогда и только тогда, когда для каждой 2-составной метки ($<x_0,v_0>$, $<x_m,v_m>$), которая удовлетворяет всем ограничениям на x_0 и x_m, существует метка для каждой переменной из x_1, …, x_{m-1} такая, что каждое бинарное ограничение на смежных переменных в этом пути удовлетворяется.

$$\forall\, csp((Z,D,C)) : \forall x_0, x_1, \ldots, x_m \in Z :$$
$$PC((x_0, x_1, \ldots, x_m), (Z,D,C)) \equiv$$
$$\equiv (\forall v_0 \in D_{x_0}, v_m \in D_{x_m} :$$
$$(satisfies((< x_0, v_0 >), C_{x_0}) \wedge$$
$$\wedge satisfies((< x_m, v_m >), C_{x_m}) \wedge$$
$$\wedge satisfies((< x_0, v_0 >, < x_m, v_m >), C_{x_0,x_m}) \Rightarrow$$
$$\Rightarrow (\exists v_1 \in D_{x_1}, v_2 \in D_{x_2}, \ldots, v_{m-1} \in D_{x_{m-1}} :$$
$$satisfies((< x_1, v_1 >), C_{x_1}) \wedge \ldots \wedge$$
$$\wedge satisfies((< x_{m-1}, v_{m-1} >), C_{x_{m-1}}) \wedge$$
$$\wedge satisfies((< x_0, v_0 >, < x_1, v_1 >), C_{x_0,x_1}) \wedge$$
$$\wedge satisfies((< x_1, v_1 >, < x_2, v_2 >), C_{x_1,x_2}) \wedge \ldots \wedge$$
$$\wedge satisfies((< x_{m-1}, v_{m-1} >, < x_m, v_m >), C_{x_{m-1},x_m}))))).$$

Определение 4. ЗУО называется **совместной по путям** тогда и только тогда, когда каждый путь в графе этой ЗУО является совместным по пути.

Это предполагает, что если ЗУО является совместной по путям, то для всех переменных x и y, всякий раз, когда составная метка (<x, a>, <y, b>) удовлетворяет ограничениям на x и y, существует метка (<z, c>) для каждой переменной z такая, что (<x, a>, <y, b>, <z, x>) удовлетворяет всем ограничениям на x, y и z.

Определение 5. Графом ограничений задачи ЗУО (Z, D, C) называется неориентированный граф, в котором каждая вершина соответствует переменной из Z, а ребра между вершинами заданы для тех пар переменных их Z, которые являются частью k-ограничения из C.

$$\forall graph((V,E)):$$
$$(V,E) = G((Z,D,C)) \equiv$$
$$\equiv ((V = Z) \wedge E = \{(x,y) \mid x,y \in Z \wedge (\exists C_S \in C : x,y \in S)\}).$$

2. Процедура достижения совместности элементов гиперграфа ограничений в ЗУО

Достижение совместности в вершинах ГО ЗУО заключается в том, чтобы для каждой переменной сократить область определения таким образом, чтобы выполнялись все унарные ограничения над данной переменной.

Описанная подзадача может быть реализована в форме процедуры (алгоритм NC-1). Ниже, на рис. 1, представлен псевдокод, который реализует достижение совместности в дугах для ЗУО (Z, D, C).

После завершения работы алгоритма NC-1, исходная задача сводится к эквивалентной задаче, для которой выполняется условие совместности в вершинах. Это достигается за счет того, что из области определения каждой переменной удаляются значения, которые нарушают унарные ограничения на

данную переменную. Если области определения представлены в виде функций, то алгоритм выполняет модификацию этих функций.

```
Алгоритм NC-1(Z, D, C )
Начало:
        Для каждой переменной x из Z выполнять:
          Для каждого значения v из Dx выполнять:
                Если <x, v> не удовлетворяет Cx, то:
                                Dx := Dx – {v};
                Конец «если».
          Конец «для».
        Конец «для».
        Возврат (Z, D, C);
Конец.
```

Рис. 1. Процедура определения совместности вершин ГО ЗУО

Временная сложность процедуры определения совместности ГО ЗУО определяется следующим образом: допустим, максимальный размер области переменной равен величине a, а количество переменных в задаче равно числу n. Так как процедура проверяет каждую переменную, временная сложность алгоритма NC-1 равна O(an).

Процедура достижения совместности в дугах ГО ЗУО. Достижение совместности в дугах может существеннее сократить область поиска решений исходной задачи по сравнению с достижением совместности в вершинах.

На рис. 2 рассматриваемая процедура представлена псевдокодом алгоритма AC-1.

```
Алгоритм AC-1(Z, D, C )
Начало:
  Выполнить подпрограмму NC-1(Z, D, C );
  // Параметр D возможно будет обновлен после
    выполнения NC-1

  Q := { <x, y> | Cx,y ∈ C };
  // <x, y> представляет дугу;
  // здесь Cy,x – то же самое, что и Cx,y

  Повторять:
    изменено := ложь;
    Для каждого <x, y> из Q, выполнять:
          // коррекция области определения Dx
          Для каждого a ∈ Dx, выполнять:
            Если не существует b ∈ Dy, такого
            чтобы (<x, a>, <y, b>) удовлетворяло Cx,y, то
                                Dx := Dx – {a};
                                Изменено := истина;
            Конец «Если»;
          Конец «Для»;
        Конец «Для»;
    пока (изменено = истина);
Конец.
```

Рис. 2. Процедура совместности дуг ГО ЗУО

Переменная Q представляет множество бинарных ограничений, подлежащих проверке. В этом множестве пары переменных, которые участвуют в ограничении, упорядочены. Другими словами, если $C_{x,y}$, то в Q помещается как пара <x, y>, так и пара <y, x>.

Алгоритм AC-1 проверяет каждую пару <x, y> из Q, и удаляет из области определения D_x те значения, которые не удовлетворяют ограничению $C_{x,y}$. Если хотя бы одно значение было удалено, цикл повторяется заново.

Постусловие алгоритма AC-1 несколько больше, чем просто совместность в дугах. Фактически, этот алгоритм достигает совместность в вершинах и совместность в дугах (другими словами, алгоритм AC-1 достигает строгую 2-совместность).

Временная сложность алгоритма для наихудшего случая выражается как $O(a^3ne)$.

Для хранения ЗУО в памяти, требуется объем памяти O(na) для хранения меток и объем памяти O(e) для хранения ограничений. Таким образом, пространственная сложность алгоритма AC-1 определяется выражением O(e + na). В случае если ограничения представлены в виде составных меток, то для хранения ограничений в худшем случае потребуется объем памяти, задаваемый выражением $O(n^2a^2)$. Алгоритм AC-1 является недостаточно эффективным, т.к. после каждого удаления любого значения из какой либо области определения, приходится перепроверять заново каждую область определения. Эффективность данного алгоритма может быть повышена за счет того, чтобы пересматривать только те бинарные ограничения, которые были затронуты удалением значения из конкретной области определения. Указанное обстоятельство определяет целесообразность в данном случае усовершенствованной процедуры на основе алгоритма AC-3.

Ниже, на рис.3, представлен псевдокод алгоритма AC-3.

```
Алгоритм AC-3(Z, D, C )
Начало:
        Выполнить подпрограмму NC-1(Z, D, C );
        // Параметр D возможно будет обновлен после вы-
полнения NC-1

        Q := { <x, y> | Cx,y ∈ C };
        // <x, y> представляет дугу;
        // здесь Cy,x – то же самое, что и Cx,y

   Пока Q ≠ {}, повторять:
          Взять произвольный элемент <x, y> из Q;
                Удалить элемент <x, y> из Q;

                // коррекция области определения Dx
                изменено := ложь;
                Для каждого a ∈ Dx, выполнять:
                  Если не существует b ∈ Dy, такого чтобы
                   (<x, a>, <y, b>) удовлетворяло Cx,y, то
                          Dx := Dx – {a};
                          Изменено := истина;
                  Конец «Если»;
              Конец «Для»;
        Если изменено = истина, то:
          Q := Q ∪ {<z,x> | Czx ∈ C ∧ z ≠ x ∧ z ≠ y};
        Конец «Если»;
   Конец «пока»;
Конец.
```

Рис. 3 . Процедура совместности дуг ГО ЗУО на основе алгоритма AC-3

Вычислительная сложность алгоритма АС-3 рассчитывается следующим образом: нижняя граница временной сложности алгоритма АС-3 определяется выражением $\Omega(a^2e)$. Верхняя оценка временной сложности задается выражением $O(a^3e)$. Пространственная сложность алгоритма аналогична алгоритму АС-1.

Процедура достижения совместности элементов ГО ЗУО. Алгоритмы достижения совместности по путям позволяют не только удалить лишние значения из областей допустимых значений переменных, но и удалить лишние составные метки из ограничений. Для таких алгоритмов ограничения должны быть представлены в виде множеств 2-составных меток.

Простейшим из алгоритмов совместности по путям является алгоритм РС-1. Псевдокод данного алгоритма приведен на рис.4 .

```
Алгоритм PC-1( Z, D, C )
Начало:
        n := |Z|;
        Y^n := C;
        Повторять:
          Y^0 := Y^n;
          Для k от 1 до n выполнять:
           Для i от 1 до n выполнять:
                Для j от 1 до n выполнять:
                  Y_ij^k := Y_ij^k-1 /\ Y_ik^k-1 * Y_kk^k-1 * Y_kj^k-1
                Конец «для»;
             Конец «для»;
          Конец «для»;
        пока Y^0 ≠ Y^n;
        C := Y^n;
        Возврат (Z, D, C);
Конец.
```

Рис. 4. Псевдокод алгоритма РС-1

3. Алгоритмы поиска решений задач программирования в ограничениях на основе недоопределенных моделей

Алгоритм локального поиска Min-conflicts. предусматривает выбор начального состояния либо случайным образом, либо с помощью жадного процесса присваивания, в котором выбирается значение для каждой переменной с минимальными конфликтами для каждой переменной по очереди. Функция leastConflicts вычисляет количество ограничений, нарушенных данным конкретным значением, с учетом остальной части текущего присваивания [4]. Псевдокод алгоритма Min-conflicts приведен на рис.5.

Алгоритм поиска с возвратом - это общий метод нахождения решения задачи, в которой требуется полный перебор всех возможных вариантов в некотором множестве М.

Решение задачи методом поиска с возвратом сводится к последовательному расширению частичного решения. Если на очередном шаге такое расширение провести не удается, то возвращаются к более короткому частичному решению и продолжают поиск дальше. Данный алгоритм позволяет найти все решения поставленной задачи, если они существуют. Для ускорения метода стараются вычисления организовать таким образом, чтобы как можно раньше выявлять заведомо неподходящие варианты. Зачастую это позволяет значительно уменьшить время нахождения решения.

```
function MIN_CONFLICTS(csp,max_steps)
возвращает решение current или индикатор неудачи failure

inpute: csp, определение задачи удовлетворения ограничений
          max_steps, количество шагов, которые разрешается
          выполнить, прежде чем отказаться от дальнейших попыток

current :=complete assignment<- первоначальное полное присваивание для csp

for 1..max_steps do:
        если current это решение: возврат current
        var := nextVar() // выбранная случайным образом конфликтующая
                         // переменная из Variables[csp]
        value := leastConflicts(var)
        var := value in initial // значение, которое минимизирует
                         // количество конфликтов
    возврат failure
```

Рис. 5. Псевдокод алгоритма Min-conflicts

Псевдокод алгоритма поиска с возвратом приведен на рис.6.

```
Алгоритм BT-1( UL, CL, D, C );
Начало;
    если UL = ø тогда:
        возврат CL;
    иначе:
        взять любую переменную x из UL;
        повторять:
            взять одно значение v из D_x;
            удалить значение v из D_x;
            если CL + {<x,v>} не нарушает ограничений C тогда:
                результат := BT-1(UL – {x},
                    CL + {<x,v>},
                    D, C );
                если результат != NIL тогда:
                    возврат результат;
                конец если;
            конец если;
        пока D_x ≠ ø;
        возврат NIL;
    конец если;
конец.
```

Рис. 6. Псевдокод алгоритма поиска с возвратом

Выводы

1. Обоснована актуальность задачи комплексной компьютеризации КППП в IT – компаниях.

2. Сформулирована задача синтеза комплексной информационной технологии реализации КППП на содержательном и формальном уровнях.

3. Формальная постановка задачи компьютеризации КППП сведена к задаче удовлетворения ограничений, представленных в форме гиперграфа.

4. Для анализа ограничений рассмотрены и усовершенствованы процедуры определения совместности элементов гиперграфа ограничений.

5. Информационная технология реализации КППП представлена в виде упорядоченного набора процедур анализа совместности элементов гиперграфа ограничений.

Литература

1. Нарусбаев, А.А. Введение в теорию проектных решений [Текст] / А.А. Нарусбаев. – Л.: Изд-во «Судостроение», 1976. – 221 с.

2. Эддоус, М. Методы принятия решений [Текст] / М. Эддоус, Р. Стэнсфилд. – М.: Аудит, Юнити, 1997. – 590 с.

3. Телерман, В.В. Удовлетворение ограничений в задачах математического программирования [Текст] / В.В. Телерман, Д.М. Ушаков // Вычислительные технологии. – 1998. – Т. 3, № 2. – С. 45–54.

4. Телерман, В.В. Недоопределенные модели: формализация подхода и перспективы развития [Текст] / В.В. Телерман, Д.М. Ушаков // Проблемы представления и обработки не полностью определенных знаний: сб. тр. РосНИИ Искусственного Интеллекта. – М., 1996. – С. 7 – 30.

Поступила в редакцию 3.06.2013, рассмотрена на редколлегии 17.06.2013

Рецензент: д-р техн. наук, проф., проф. кафедры «Программная инженерия» С.Ю. Шабанов-Кушнаренко, ХНУРЭ, Харьков.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КАЛЕНДАРНОГО ПЛАНУВАННЯ ПРИ УПРАВЛІННІ ПОРТФЕЛЕМ ЗАМОВЛЕНЬ ІТ-КОМПАНІЇ В УМОВАХ НЕВИЗНАЧЕНОСТІ

М.В. Потапова

Розглянуто фактори, що впливають на ефективність рішень, пов'язаних з календарним плануванням програмних проектів (КППП) в ІТ-компанії. Показана недостатня ефективність існуючого підходу до комп'ютеризації КППП на основі використання стандартних офісних додатків. Розроблено постановку задачі синтезу спеціалізованої інформаційної технології КППП, яка на формальному рівні зведена до задачі задоволення обмежень, у формі гіперграфа. Описано стандартні та модифіковані процедури аналізу гіперграфа і представлена інформаційна технологія КППП у вигляді впорядкованого набору процедур аналізу спільності елементів гіперграфа обмежень.

Ключові слова: інформаційна технологія, ІТ-компанія, гіперграф обмежень, невизначеність, завдання задоволення обмежень, програмний проект.

INFORMATION TECHNOLOGY OF SCHEDULING IN CASE OF PORTFOLIO MANAGEMENT OF ORDERS OF THE IT COMPANY IN THE CONDITIONS OF UNCERTAINTY

M.V. Potapova

The factors influencing efficiency of decisions, connected to the scheduling of program projects (SPP) in IT – company are considered. Insufficient efficiency of existing approach to computerization of SPP on the basis of use of standard office applications is shown. The problem definition of synthesis of specialized information technology of SPP which at the formal level is brought together to the task of satisfaction of restrictions, in the form of the hypergraph is developed. The standard and modified procedures of the analysis of the hypergraph are described and the information technology of SPP in the form of ordered set of procedures of the analysis of compatibility of elements of the hypergraph of restrictions is provided.

Keywords: information technology, IT-company, the hypergraph of restrictions, uncertainty, the task of satisfaction of restrictions, the program project.

Потапова Марина Викторовна – зав. лаб. каф. финансов Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: marina-k604@yandex.ru.