

КАЛЕНДАРНОЕ ПЛАНИРОВАНИЕ ДЛЯ УНИФИЦИРОВАННОЙ МОДЕЛИ ОБЪЕКТА УПРАВЛЕНИЯ ДИСКРЕТНОГО ПРОИЗВОДСТВА

1. Введение

Календарное планирование при управлении реализацией производственного плана (производственного заказа, изделия, узла, детали) является важнейшей компонентой любой производственной системы. Несмотря на кажущуюся очевидность задачи, реализация календарного планирования наталкивается на множество проблем. Достаточно сказать, что ни одна из находящихся на рынке отечественных программных продуктов ERP-систем (или систем, приближенных к этой планке) не имеет в своем составе компоненты Shop Flow Control (диспетчирование/управление производством). Выпущенная в сентябре 2002 года корпорацией «Галактика» (нашими соседями из ближнего зарубежья, один из лидеров-производителей ERP-систем) коммерческая версия системы "Галактика-Производство" так и не имеет в своем составе модулей календарного планирования производства и контроллинга за ходом выполнения производственного процесса.

План производства, производственный заказ и, как частный случай производственного заказа, производственная партия изделий являются объектами управления в системе управления производственными процессами. Если рассматривать реализацию производственного заказа как реализацию конкретного проекта, то логично предложить к использованию как средство календарного планирования и управления проектом методы сетевого планирования и управления, имеющие наиболее разработанную математическую теорию [1]. Но именно сложность в реализации этих методов и большие трудозатраты при подготовке информации для использования методов сетевого планирования и управления являются наиболее проблемными при попытках корректной реализации календарного планирования. Сложность эта возрастает на два порядка при необходимости решения задачи минимизации времени выполнения проекта (плана, заказа и т.п.) при ограниченных ресурсах. Крупные интегрированные производственные системы (типа SAP/R3, Ваап и т.п.), где проблема календарного планирования решена корректно и в полном объеме, не могут служить эталоном по причине большой стоимости систем. И как следствие этого – невозможность массового использования этих систем. Но даже в таких системах существует статус производственного заказа "окончательно запланирован" (как, например, в Ваап), что исключает возможность перепланирования такого заказа в критических и аномальных ситуациях, что для нашего рынка является скорее правилом, чем исключением.

Малые и средние интегрированные производственные системы явно или не столь явно используют различные ограничения и упрощения для получения конечного результата в компоненте календарного планирования. Можно привести наиболее типовые упрощения, используемые для этих целей:

– расчет и перерасчет календарных временных параметров производится в специализированном программном продукте типа MS Project. Система (например, продукт

швейцарской фирмы OKS) подготавливает исходные данные для использования в специализированном продукте. Налицо использование чужеродного для исходной системы продукта (как следствие – удорожание исходной системы), большие организационные проблемы для подготовки и перекачки информации из системы в систему, наличие некоторой промежуточной фазы обработки;

- расчет календарных сроков производственных заказов без учета ограничений на ресурсы (фактически расчет одного заказа производится вне зависимости от расчета календарных сроков по другим заказам);

- упрощения в структуре производственного заказа, когда заказ сводится к планированию выпуска какого-то количества продукции одного вида, при этом зачастую технологию производства продукции сводят к простейшей линейной цепочке технологических операций.

Оставим за рамками рассмотрения настоящей статьи временные процессы технической и материальной подготовки производственного заказа к выполнению и сконцентрируем наше внимание на производственном процессе изготовления продукции по заказу. Если исходить из предположения, что наиболее корректно управлять проектом (производственным заказом) мы можем, используя методы сетевого планирования и управления [2], то задача календарного планирования и управления производственным заказом сводится к составлению сетевого графика производственного процесса, его расчету и управлению сетевым графиком в процессе производства.

2. Описание методики автоматического формирования сетевого графика

Классический сетевой график (рис. 1) представляет собой антисимметричный ориентированный граф с одной входной вершиной и одной выходной вершиной [3].

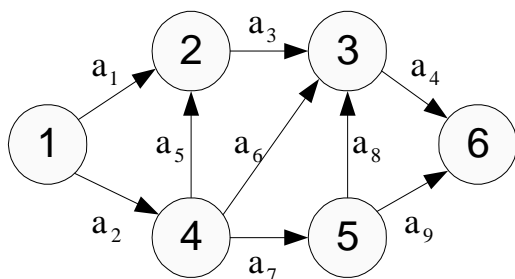


Рис. 1. Вид сетевого графика

Взаимная обусловленность работ сетевого графика определяется тем, что выполнение некоторых работ нельзя начать раньше, чем будут завершены некоторые другие, опорные работы. Физическая реализация сетевой модели представляет собой матрицу, каждая строка которой описывает параметры каждой работы сетевого графика, в том

числе и коды начальных и конечных событий работы, которые в неявном виде и формируют топологию графика.

В явном виде сетевой график изготовления изделия отсутствует в конструкторской и технологической документации, и это является самым серьезным препятствием для широкого использования методов сетевого планирования и управления в производстве. При всем при этом любая производственная система обладает самодостаточной информацией для автоматического построения сетевого графика выполнения производственного плана (заказа, изделия), его расчета, оптимизации и перепланирования.

Дальнейшее обсуждение темы будем вести, исходя из предположения, что мы имеем дело с дискретным производством (механообработка, приборостроение и т.п.).

Автором ранее предлагалась унифицированная модель объекта управления дискретного производства, основанная на предположении об иерархичности ресурсов и объектов управления дискретного производства и об использовании универсального алгоритма "разузловки",

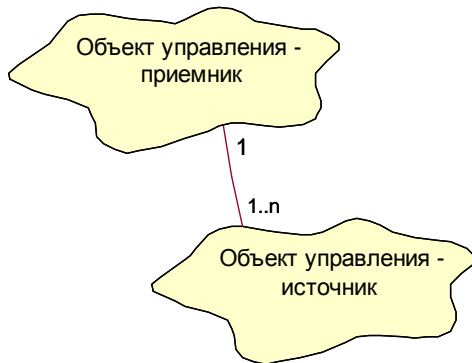


Рис. 2. Модель объекта управления

позволяющего получить это дерево объектов [4].

В такой модели отношения между объектами двух соседних уровней иерархии можно представить следующей схемой (рис. 2).

Физическая реализация такой модели представляет собой матрицу, строки которой содержат следующую основную информацию:

- обозначение объекта-приемника;
- количество объекта-приемника;
- обозначение объекта-источника;
- количество объекта-источника.

Таким способом можно описать взаимосвязь плана производства и производственных заказов, производственного заказа и изготавливаемых Item-единиц, Item-единиц с конструкторскими и технологическими компонентами для их изготовления. Унифицированный метод "разузловки" позволяет получить не только состав любого объекта управления, но и топологию дерева входимости объектов управления друг в друга, что дает возможность провести рассуждения о получении методики автоматического формирования сетевого графика изготовления любого объекта управления.

Рассмотрим методику формирования сетевого графика для изготовления производственной партии изделий. Предлагаемая методика является унифицированной и применима для формирования сетевых графиков производственных заказов и плана производства в целом.

Представим компоненты информационного потока преобразования информации как:

- $x_{изд}$ – информация об изготавливаемом изделии;
- $x_{спец}$ – информация о конструкторских спецификациях для изготавливаемого изделия;
- $x_{мтк}$ – информация о маршрутно-технологических картах для изготавливаемого изделия;
- $x_{сост}$ – информация о составе изготавливаемого изделия;
- $x_{граф}$ – сетевой график изготовления изделия.

Тогда желаемый информационный граф преобразования имеющейся информации об изготавливаемом изделии в сетевой график изготовления изделия можно было бы изобразить следующим образом (рис. 3):

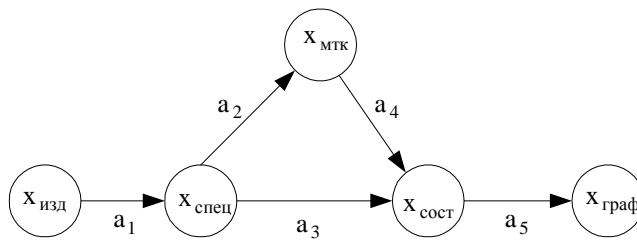


Рис. 3. Информационный граф преобразования информации

Из представленного графа наиболее неясным является преобразование a_5 – преобразование состава изделия в сетевой график изготовления. Тем не менее такое преобразование возможно. Как уже отмечалось выше, метод "разузловки" для унифицированной модели объекта управления позволяет представить иерархическую структуру объекта управления в виде линейного связанного списка компонент, обеспечивающего четкую взаимную обусловленность следования узлов и деталей. Это и понятно: сборку изделия невозможно произвести без изготовления необходимых узлов и деталей.

Именно эта взаимная обусловленность следования узлов и деталей позволяет говорить о возможности преобразования результатов «разузловки» в ориентированный граф, т.е. фактически в сетевой график изготовления изделия (узла). Задача сводится к дополнению результатов «разузловки» фиктивными работами с продолжительностью 0 для формирования входной вершины графа и разработки алгоритма нумерации вершин графа с целью обеспечения твердого соблюдения нумерации событий $i < j$ [5].

Поясним предлагаемый алгоритм преобразования результатов «разузловки» в сетевой график на примере.

Пусть имеем некоторое производимое изделие " Drawing 0", которое описывается следующей спецификацией.

Таблица 1

Drawing 0 – 1 шт.			
Номер позиции	Обозначение чертежа	Признак узла	Количество
1	Drawing 1		1
2	Drawing 2	Узел	2
3	Drawing 6		1

Узел "Drawing 2" описывается следующей спецификацией.

Таблица 2

Drawing 2 – 1 шт.			
Номер позиции	Обозначение чертежа	Признак узла	Количество
1	Drawing 3		2
2	Drawing 4	Узел	1

Узел "Drawing 4" описывается следующей спецификацией.

Таблица 3

Drawing 2 – 1 шт.			
Номер позиции	Обозначение чертежа	Признак узла	Количество
1	Drawing 5		2
2	Drawing 7		5

Тогда, после отработки алгоритма «разузловки», имеем следующий результат «разузловки» изделия Drawing 0 в количестве 1.

Таблица 4

№	Уровень	Обозначение чертежа	Узел	Количество
1	0	Drawing 0	Узел	1
2	1	Drawing 1		1
3	1	Drawing 2	Узел	2
4	2	Drawing 3		4
5	2	Drawing 4	Узел	2
6	3	Drawing 5		4
7	3	Drawing 7		10
8	1	Drawing 6		1

Графически дерево «разузловки» будет выглядеть так (рис. 4):

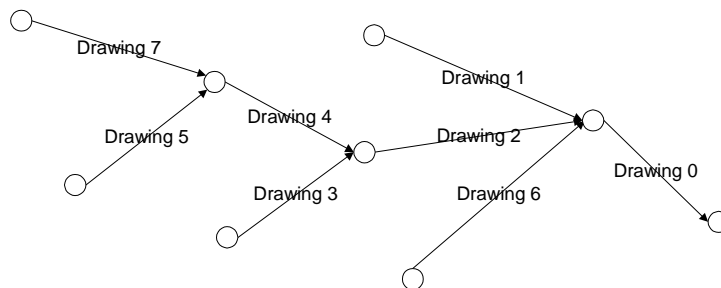


Рис. 4. Результат «разузловки»

Предлагается следующий алгоритм составления сетевого графика на основе матрицы результатов «разузловки»:

1. Начальное событие графика устанавливается равным некоему целому числу I .
2. Конечное событие графика устанавливается равным некоему целому числу J , причем J много больше I ($J \gg I$, $J - I$ – число, превышающее общее число строк в результирующей матрице «разузловки»).
3. Позиции «разузловки», которые не имеют признака узла, должны соединяться фиктивной работой с продолжительностью 0 с начальным событием I .

4. Конечному событию работы уровня 0 устанавливаем значение J , начальному событию работы уровня 0 – значение $J - 1$, присваиваем $J_T = J - 1$.

5. Устанавливаем $I_T = 1$, начальным событиям работ без уровней (фиктивным работам) устанавливаем значение 1, конечным событиям работ без уровней (фиктивным работам) – значение $I_T + 1$ (с наращиванием I_T после каждого присваивания, по числу фиктивных работ).

6. Находим работы, следующие за фиктивными работами, и устанавливаем начальное событие таких работ, равным конечному событию фиктивных работ.

7. Находим работы, входящие в работу 0-го уровня, и устанавливаем конечное событие для таких работ, равным $J_T = J - 1$.

8. Дальнейшая работа с матрицей в цикле. Первый шаг цикла: делаем проход по таблице, начиная сверху; для тех строчек, где не заполнен столбец "Нач. событие", устанавливаем в них значение на 1 меньше от максимального текущего на первом шаге J_T (J_T , на первом шаге $J - 1$); если строка "узел", проставляем значение "Кон. событие", равное J_T , в строках с номером уровня на уровень больший, чем текущий номер уровня (на первом шаге $J - 1$).

9. Делаем проход по таблице, если встретились строки с незаполненными столбцами "Нач. событие" повторяем шаг 8 алгоритма.

10. По завершении обработки таблицы разузловки будем иметь автоматически установленные номера конечных и начальных событий работ при твердо соблюденном соотношении номеров начальных и конечных событий $i < j$.

Проиллюстрируем предлагаемый алгоритм на приведенном выше примере. После третьего шага алгоритма вид сетевого графика будет следующий (рис. 5):

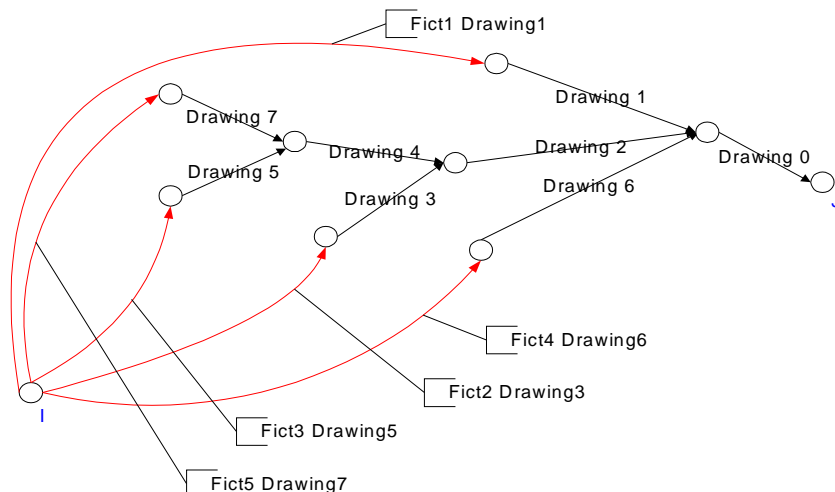


Рис. 5. Преобразование дерева «разузловки» в ориентированный граф

Таблица «разузловки» дополняется дополнительными строчками.

Таблица 5

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1		
2	1	Drawing 1		1		

3	1	Drawing 2	Узел	2		
4	2	Drawing 3		4		
5	2	Drawing 4	Узел	2		
6	3	Drawing 5		4		
7	3	Drawing 7		10		
8	1	Drawing 6		1		
9		Fict1 Drawing 1		1		
10		Fict2 Drawing 3		1		
11		Fict3 Drawing 5		1		
12		Fict4 Drawing 6		1		
13		Fict5 Drawing 7		1		

С учетом правил изложенных выше, можем установить следующие номера событий.

Таблица 6

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1		J
2	1	Drawing 1		1		
3	1	Drawing 2	Узел	2		
4	2	Drawing 3		4		
5	2	Drawing 4	Узел	2		
6	3	Drawing 5		4		
7	3	Drawing 7		10		
8	1	Drawing 6		1		
9		Fict1 Drawing 1		1	I	
10		Fict2 Drawing 3		1	I	
11		Fict3 Drawing 5		1	I	
12		Fict4 Drawing 6		1	I	
13		Fict5 Drawing 7		1	I	

Для нулевого уровня начальное событие устанавливаем на 1 меньше от конечного; для работ, выходящих из 0 события, как работ исходящих из вершины 0-го ранга, нумерацию задаем через 1.

Таблица 7

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1	J-1	J
2	1	Drawing 1		1		
3	1	Drawing 2	Узел	2		
4	2	Drawing 3		4		
5	2	Drawing 4	Узел	2		
6	3	Drawing 5		4		
7	3	Drawing 7		10		
8	1	Drawing 6		1		
9		Fict1 Drawing 1		1	I	I+1
10		Fict2 Drawing 3		1	I	I+2

11		Fict3 Drawing 5		1	I	I+3
12		Fict4 Drawing 6		1	I	I+4
13		Fict5 Drawing 7		1	I	I+5

Затем устанавливаем конечные события для работ 1-го уровня и начальные события для работ, начинающихся за фиктивными работами.

Таблица 8

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1	J-1	J
2	1	Drawing 1		1	I+1	J-1
3	1	Drawing 2	Узел	2		J-1
4	2	Drawing 3		4	I+2	
5	2	Drawing 4	Узел	2		
6	3	Drawing 5		4	I+3	
7	3	Drawing 7		10	I+5	
8	1	Drawing 6		1	I+4	J-1
9		Fict1 Drawing 1		1	I	I+1
10		Fict2 Drawing 3		1	I	I+2
11		Fict3 Drawing 5		1	I	I+3
12		Fict4 Drawing 6		1	I	I+4
13		Fict5 Drawing 7		1	I	I+5

Таблица 9

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1	J-1	J
2	1	Drawing 1		1	I+1	J-1
3	1	Drawing 2	Узел	2	J-2	J-1
4	2	Drawing 3		4	I+2	J-2
5	2	Drawing 4	Узел	2		J-2
6	3	Drawing 5		4	I+3	
7	3	Drawing 7		10	I+5	
8	1	Drawing 6		1	I+4	J-1
9		Fict1 Drawing 1		1	I	I+1
10		Fict2 Drawing 3		1	I	I+2
11		Fict3 Drawing 5		1	I	I+3
12		Fict4 Drawing 6		1	I	I+4
13		Fict5 Drawing 7		1	I	I+5

Эта работа циклическая; делаем так, пока не обработаем всю таблицу.

Таблица 10

№	Уровень	Обозначение чертежа	Узел	Количество	Нач. событие	Кон. событие
1	0	Drawing 0	Узел	1	J-1	J
2	1	Drawing 1		1	I+1	J-1

3	1	Drawing 2	Узел	2	J-2	J-1
4	2	Drawing 3		4	I+2	J-2
5	2	Drawing 4	Узел	2	J-3	J-2
6	3	Drawing 5		4	I+3	J-3
7	3	Drawing 7		10	I+5	J-3
8	1	Drawing 6		1	I+4	J-1
9		Fict1 Drawing 1		1	I	I+1
10		Fict2 Drawing 3		1	I	I+2
11		Fict3 Drawing 5		1	I	I+3
12		Fict4 Drawing 6		1	I	I+4
13		Fict5 Drawing 7		1	I	I+5

Работа по автоматической установке номеров начальных и конечных событий завершена (при твердом соблюдении $i < j$).

Фактически мы имеем автоматически сформированный сетевой график на основе «разузловки».

Реально мы имеем дело с укрупненным сетевым графиком, так как за каждым обозначением чертежа (не фиктивным) стоит линейная последовательность технологических операций, определенная маршрутно-технологической картой для данной компоненты. Легко видеть, что расширить уже составленный сетевой график до уровня технологических операций из маршрутно-технологических карт не составит труда. Аналогичными рассуждениями возможно строить сетевые графики исполнения производственного заказа как совокупности изготавливаемых партий изделий и производственного плана как совокупности производственных заказов.

В любом случае алгоритмически мы получаем сетевой график, представленный списком своих дуг – работ (P_i, P_j) и их продолжительностей, причем алгоритмом нумерации событий для любой дуги (P_i, P_j) имеет место $i < j$, и для каждой работы продолжительность удовлетворяет неравенству $t_{ij} \geq 0$.

Таблица 11

Работа (P_i, P_j)		Продолжительность t_{ij}
Начальное событие P_i	Конечное событие P_j	
P_0	P_s	t_{os}
.	.	.
.	.	.
P_i	P_j	t_{ij}
.	.	.
.	.	.

3. Выводы

Основная предлагаемая идея состоит в разработке унифицированной методики, позволяющей на базе информации унифицированной модели объекта управления через алгоритмы "разузловки" и преобразования дерева "разузловки" в ориентированный граф автоматически формировать сетевой график выполнения производственного плана (производственного заказа, производственной партии изделий).

Имея корректно автоматически построенный сетевой график выполнения производственного плана (заказа, изделия, узла, детали) и используя уже широко известные алгоритмы и методы сетевого планирования, можно получить все параметры графика (при различных вариантах расчета как "от начала", так и "от конца"): критическое время и критический путь, минимальные и максимальные времена наступления событий, резервы времени [5, 6]. Более того, для сетевых графиков производственных планов и заказов можно провести оптимальное по времени распределение ограниченных ресурсов и в этом распределении будут участвовать все производственные заказы (изделия), имеющие статус "в производстве".

Из подсистемы управления изготовлением партий деталей и узлов постоянно будет происходить "подпитка" информацией о выполнении тех или иных технологических операций (полностью или частично по фактически использованному времени), а это означает, что уже сформированный сетевой график в любой момент времени готов к перерасчету и перераспределению ресурсов. Также понятно, что уже сформированный и рассчитанный сетевой график производственного плана легко дополнить включениями сетевых графиков новых производственных заказов, и возможные изменения в календарных сроках могут быть оперативно учтены перерасчетом откорректированного графика.

Практическая значимость предлагаемой методики очевидна для производства – информация унифицированной модели объекта управления является самодостаточной для автоматического построения сетевого графика изготовления изделий.

СПИСОК ЛИТЕРАТУРЫ

1. Куперштейн В.И. Современные информационные технологии в делопроизводстве и управлении. – Санкт-Петербург: БХВ, 1999. – 388 с.
2. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. – Москва: Мир, 1984. – 496 с.
3. Кристофидес Н. Теория графов. – Москва: Мир, 1978. – 432 с.
4. Каплунов М.И. Унифицированная модель объекта управления дискретного производства // Математические машины и системы. – 2003. – № 2.
5. Зуховицкий С.И., Радчик И.А. Математические методы сетевого планирования. – Москва: Наука, 1965. – 296 с.
6. Голенко Д.И. Статистические методы сетевого планирования и управления. – Москва: Наука, 1968. – 378 с.