

Реализация сетевых методов планирования и управления**Дата:** 17/03/2006**Номер:** (05) УЭКС, 1/2006

Аннотация: Принципиально любая задача сетевого планирования может быть сведена к решению задачи по теории графов. Анализ пакетов символьной математики на предмет использования для решения задач сетевого планирования показал, что наиболее оптимальным является наличие модулей по теории графов. При этом достаточно только четко и грамотно провести постановку задачи, подготовить данные и обработать их с использованием стандартных процедур.

Abstract: Essentially any problem of network planning can be shown to the problem decision under the theory of counts. The analysis of packages of symbolical mathematics about use for the decision of problems of network planning has shown, that the optimal is presence of modules under the theory of counts. Thus it is enough to spend accurately and competently only problem statement, to prepare data and to process them with use of standard procedures.

Ключевые слова: сетевое планирование, сетевой график, символьная математика

Keywords: network planning, network schedule, symbolical mathematics

В.Л.Шарстнёв

кандидат технических наук, доцент

Е.Ю. Вардомацкая

Республика Беларусь, Витебский государственный технологический университет

Сетевое планирование – это комплекс графических и расчетных методов организационных мероприятий, обеспечивающих моделирование, анализ и динамическую перестройку плана выполнения сложных проектов и разработок [1]. Характерной особенностью таких проектов является то, что они состоят из ряда отдельных элементарных работ. Они обуславливают друг друга так, что выполнение некоторых работ не может быть начато раньше, чем завершены некоторые другие.

Классическое представление задачи сетевого планирования показано на рис.1. Основными понятиями сетевых моделей являются понятия работы, события и пути [2].

Работа – это некоторый процесс, имеющий протяженность во времени, требующий затрат каких-либо ресурсов и приводящий к достижению определенного результата (обозначаются стрелками). Событие – это момент времени, когда завершаются одни работы и начинаются другие (обозначается кружком, см. рис. 2).

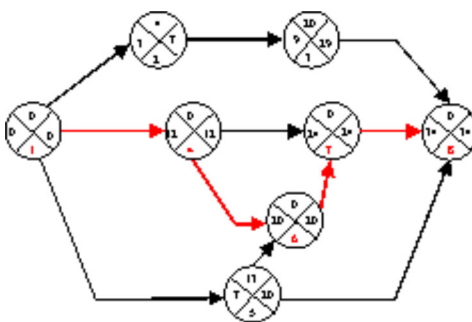


Рис.1

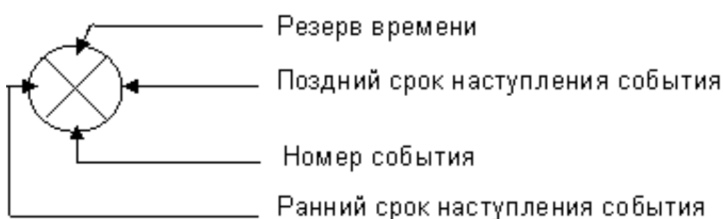
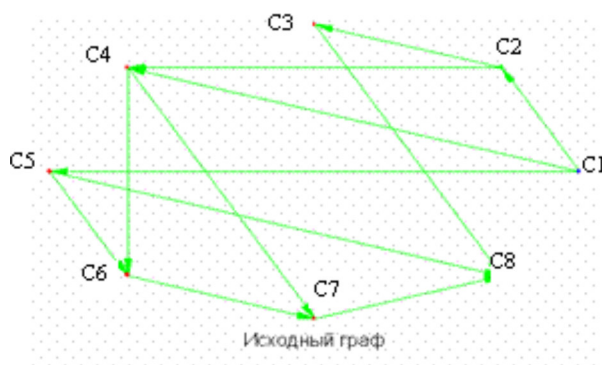


Рис.2

Путь – это любая последовательность работ в сетевом графике, в которой конечное событие одной работы совпадает с начальным событием следующей за ней работы. Различают:

- полный путь – путь от исходного до завершающего события;
- критический путь – максимальный по продолжительности полный путь;
- подкритический путь – полный путь, ближайший по длительности к критическому пути.



Алгоритм расчета сетевого графика общеизвестен и включает в себя расчет ранних и поздних сроков наступления событий, резервов времени каждого события и затем определение критического пути [3].

В рассматриваемом примере критический путь проходит через события 1 – 4 – 6 – 7 – 8 и его длительность равна 34 дням.

Современные информационные технологии позволяют использовать некоторые другие подходы в решении приведенного выше типа задач. Можно достаточно легко и просто, опираясь на пакеты символьной математики, получить наглядное и точное решение. При этом совершенно не обязательно владеть алгоритмом решения.

Достаточно четко и грамотно провести постановку задачи, подготовить данные и обработать их с использованием стандартных процедур [4].

Принципиально любая задача сетевого планирования может быть сведена к решению задачи по теории графов. Анализ пакетов символьной математики [5] на предмет использования для решения задач сетевого планирования показал, что наиболее оптимальным является наличие модулей по теории графов.

Покажем, как стандартную задачу сетевого планирования решить с использованием системы символьной математики MuPAD Pro 3.1 (<http://www.sciface.com/>) [5].

1. Зададим вершины (события) Vert исходного графа

```
• Vert:=[c1, c2, c3, c4, c5, c6, c7, c8]
```

2. Зададим ребра (работы) Ed исходного графа

```
• Ed:=[[c1, c2], [c1, c4], [c1, c5], [c2, c4], [c4, c6], [c5, c6], [c4, c7],  
[c6, c7], [c2, c3], [c3, c8], [c7, c8], [c5, c8]]
```

3. Зададим размер ребер (продолжительность работ) EdW исходного графа

```
• EdW:=[3,12,7,5,8,0,10,4,6,5,10,10]
```

4. Зададим необязательное описание ребер (работ)

```
• EdD:=["Работа 1-2", "Работа 1-4",  
"Работа 1-5", "Работа 2-4", "Работа 4-6", "Работа 5-6", "Работа 4-7",  
"Работа 6-7", "Работа 2-3", "Работа 3-8", "Работа 7-8", "Работа 5-8"]
```

5. Сформируем исходный направленный граф (ключевое слово – Directed)

```
• G := Graph(Vert, Ed, EdgeWeights = EdW, EdgeDescriptions = EdD, Directed):
```

6. Прорисуем исходный граф

```
plot(Graph::plotCircleGraph(G, EdgeColor = RGB::Green), Footer="Исходный граф")
```

7. При необходимости вывод информации об исходном графе

```
• Graph::printGraphInformation(G):
```

8. Определим все возможные критические пути относительно исходного события с 1

```
• Temp:= Graph::longestPath(G, c1, Length, Path)
```

Очень информативный вывод данных позволяет сделать заключение о том, длительность критического пути “с 1 – с 8” составляет 34 дня. События, относящиеся к критическому пути, представляют собой последовательность “с1 – с4 – с6 – с7 – с8”

.

Выделим значение длительности критического пути

```

• CPV:=CriticalPathValue:=op(Temp[1],nops(Temp[1]))
  c8 = 34

```

и события, относящиеся к критическому пути

```

• CP:=CriticalPath:=op(Temp[2],nops(Temp[2]))
  c8 = [c8, c7, c6, c4, c1]

```

Принципиально задача об отыскании критического пути решена. Однако, с целью наглядности, покажем, как график критического пути отображается на исходном графе.

9. Создадим матрицу событий, относящихся к критическому пути

```

• x:=matrix(nops(CP[2])-1,2):
• for i from nops(CP[2])-1 downto 1 do
  for j from 1 to 2 do
    if j=1 then x[i,j]:=op(CP[2],i+1)
    else x[i,j]:=op(CP[2],i)
  end_if:
end_for:
end_for:
• x

```

10. Преобразуем матрицу событий в список, необходимый для использования в процедурах прорисовки

```

• kk:=coerce(x,DOM_LIST)

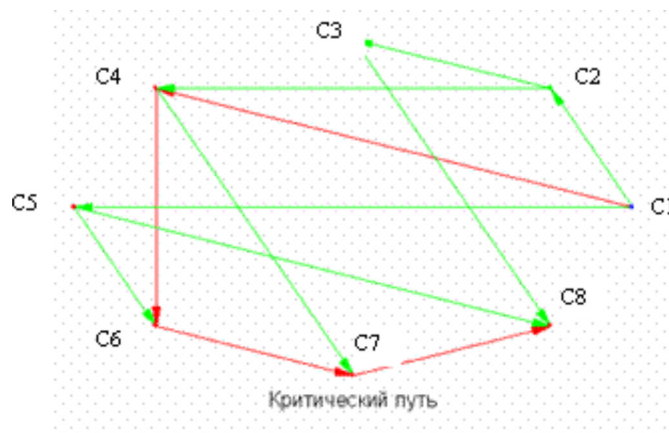
```

11. Осуществим окончательный вывод графического представления решения задачи сетевого планирования

```

• G := Graph(Vert, Ed, EdgeWeights = EdW, EdgeDescriptions = EdD, Directed):
  plot(Graph::plotCircleGraph(G,EdgeColor = RGB::Green, SpecialEdges = kk,
    SpecialEdgeColor = RGB::Red), Footer="Критический путь")

```



Предлагаемый алгоритм позволяет сформировать отдельную библиотечную процедуру, позволяющую решать задачи сетевого планирования в разрезе отыскания критического пути. Дальнейшим этапом развития алгоритма является получение возможности оптимизации производственной программы по временным и

стоимостным параметрам [6]. Программная реализация данного алгоритма внедрена и используется в учебном процессе на экономическом факультете Витебского государственного технологического университета в курсах: экономико-математические методы и модели, компьютерные информационные технологии.

Литература

1. Абчук В.А. Экономико-математические методы. Элементарная математика и логика. Методы исследования операций. – СПб.: Союз, 1999. – 320 с.
2. Похабов В.И. Экономико-математические методы и модели (практикум). – Мн.: БНТУ, 2003. – 130 с.
3. Бажин И.И. Информационные системы менеджмента. – М.: ГУ-ВШЭ, 2000. – 688 с.
4. Юферева О.Д. Экономико-математические методы. Мн.: БГЭУ, 2002. – 56 с.
5. Официальный сайт MuPAD . <http://www.sciface.com/>
6. Костевич Л.С. Информационные технологии оптимальных решений и реинжиниринг в повышении эффективности менеджмента. Научное издание Мн.: БГЭУ, 2000. – 108 с.

Это статья Электронный научный журнал "Управление экономическими системами"
<http://uecs.mcnip.ru>

URL этой статьи:

<http://uecs.mcnip.ru/modules.php?name=News&file=article&sid=30>