

ОРГАНІЗАЦІЯ ВИРОБНИЦТВА ТА ДОСЛІДЖЕННЯ ОПЕРАЦІЙ

ПРОВЕРКА ПРАВИЛ ЗАПУСКА В ГРУППОВОЕ ПРОИЗВОДСТВО НА ОСНОВЕ СТАТИСТИЧЕСКОГО ЭКСПЕРИМЕНТА

Алексей Владимирович Антонюк

соискатель кафедры менеджмента

Восточноукраинского национального университета им. В. Даля,
студент специальности "Математика"

Курского педагогического университета

В соответствии с повышающимся интересом к производительной – как с технологической точки зрения, так и с управленческой – организации операционных процессов, актуальность приобретает рациональное упорядочение запуска предметов труда в обработку в условиях группового производства. А. Петровым были выявлены четыре правила упорядочения, но это не конечное число возможных правил, строя календарные графики можно разглядеть новые варианты упорядочения, при этом возникает вопрос: а существует ли конечное число правил?

Для случая обработки деталей с одинаковыми технологическими маршрутами при любом числе операций, разработкой правил упорядочения занимался Петров В. А. в книге "Групповое производство и автоматизированное оперативное управление". Однако в ходе исследования условий применения правил запуска была поставлена задача проверки частоты преобладания того или иного правила. Основой такой проверки было выбрано проведение статистического эксперимента по выявлению правила предпочтительности при определении упорядоченной очередности запуска предметов труда, наиболее часто определяющего минимальную совокупную длительность цикла.

В нашем распоряжении имелась программа имитационного моделирования, посредством которой, был проведен статистический эксперимент. Для этого потребовалось записать алгоритмы правил на языке данной программы. Статистические эксперименты проходили на примерах технологического цикла из четырех операций, которые проходит шесть предметов труда. Пример таких данных представлен в табл. 1. Для ввода второго правила Петрова, необходимо ввести следующие величины: $a_{11}=t_{21}+t_{11}$, $b_{12}=t_{22}+t_{12}$, $c_{13}=t_{23}+t_{13}$, $d_{14}=t_{24}+t_{14}$, $e_{15}=t_{25}+t_{15}$, $f_{16}=t_{26}+t_{16}$; $a_{21}=t_{31}+t_{41}$, $b_{22}=t_{32}+t_{42}$, $c_{23}=t_{33}+t_{43}$, $d_{24}=t_{34}+t_{44}$, $e_{25}=t_{35}+t_{45}$, $f_{26}=t_{36}+t_{46}$; $a=a_{21}-a_{11}$, $b=b_{22}-b_{12}$, $c=c_{23}-c_{13}$, $d=d_{24}-d_{14}$, $e=e_{25}-e_{15}$, $f=f_{26}-f_{16}$.

a_{11} – является суммарной трудоемкостью обработки по операциям левой половины таблицы 1, a_{21} – соответственно правой половины таблицы 1. a , b , c , d , e , f – разница сумм второй и первой половины таблицы 1, по каждому автомобилю.

Таблица 1

Нормы времени деталей по операциям

	1 операция	2 операция	3 операция	4 операция
1 деталь	(t11) 10	(t21) 10	(t31) 10	(t41) 10
2 деталь	(t12) 10	(t22) 10	(t32) 10	(t42) 10
3 деталь	(t13) 10	(t23) 20	(t33) 60	(t43) 90
4 деталь	(t14) 90	(t24) 60	(t34) 20	(t44) 10
5 деталь	(t15) 20	(t25) 90	(t35) 10	(t45) 60
6 деталь	(t16) 60	(t26) 30	(t36) 90	(t46) 10

Дабы избежать повторения величин a, b, c, d, e, f, в программе были прибавлены к ним числа от 1 до 6, эта условность никак не повлияет на длину цикла, так как эта величина используется только для определения порядка запуска.

Для определения последовательности запуска по второму правилу Петрова использовались следующие функции:

Функция GetIndexBVS. Возвращает порядковый номер в неотсортированном массиве по рангу в отсортированном массиве. $y_1 = \text{GetIndexBVSD}(6, [a, b, c, d, e, f])$, 6 – ранг элемента массива, отсортированного в порядке возрастания.

Функция GetValueBI. Возвращает значение из неотсортированного массива. $t_{111} = \text{getvaluebi}(1, [t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}])$, 1 – порядковый номер числа из указанного массива.

Привожу фрагмент программы с вышеуказанными функциями.

```

y1=GetIndexBVSD (6, [a,b,c,d,e,f])
t111=getvaluebi(y1,[t11,t12,t13,t14,t15,t16])
t211=getvaluebi(y1,[t21,t22,t23,t24,t25,t26])
t311=getvaluebi(y1,[t31,t32,t33,t34,t35,t36])
t411=getvaluebi(y1,[t41,t42,t43,t44,t45,t46])
y2=GetIndexBVSD (5, [a,b,c,d,e,f])
t121=getvaluebi(y2,[t11,t12,t13,t14,t15,t16])
t221=getvaluebi(y2,[t21,t22,t23,t24,t25,t26])
t321=getvaluebi(y2,[t31,t32,t33,t34,t35,t36])
t421=getvaluebi(y2,[t41,t42,t43,t44,t45,t46])
y3=GetIndexBVSD (4, [a,b,c,d,e,f])
t131=getvaluebi(y3,[t11,t12,t13,t14,t15,t16])
t231=getvaluebi(y3,[t21,t22,t23,t24,t25,t26])
t331=getvaluebi(y3,[t31,t32,t33,t34,t35,t36])
t431=getvaluebi(y3,[t41,t42,t43,t44,t45,t46])
y4=GetIndexBVSD (3, [a,b,c,d,e,f])
t141=getvaluebi(y4,[t11,t12,t13,t14,t15,t16])
t241=getvaluebi(y4,[t21,t22,t23,t24,t25,t26])
t341=getvaluebi(y4,[t31,t32,t33,t34,t35,t36])
t441=getvaluebi(y4,[t41,t42,t43,t44,t45,t46])
y5=GetIndexBVSD (2, [a,b,c,d,e,f])
t151=getvaluebi(y5,[t11,t12,t13,t14,t15,t16])
t251=getvaluebi(y5,[t21,t22,t23,t24,t25,t26])
t351=getvaluebi(y5,[t31,t32,t33,t34,t35,t36])
t451=getvaluebi(y5,[t41,t42,t43,t44,t45,t46])

```

```

y6=GetIndexBVSD (1, [a,b,c,d,e,f])
t161=getvaluebi(y6,[t11,t12,t13,t14,t15,t16])
t261=getvaluebi(y6,[t21,t22,t23,t24,t25,t26])
t361=getvaluebi(y6,[t31,t32,t33,t34,t35,t36])
t461=getvaluebi(y6,[t41,t42,t43,t44,t45,t46]),

```

где значения t_{nmj} составляют следующие элементы: n – номер операции, m – номер автомобиля, j – порядковый номер правила запуска.

После того как порядок запуска автомобилей в ремонт определен, в программу был записан цикл прохождения автомобилей по постам, конечным числом которого является длительность цикла по заданному правилу.

t211n=t111k	t311n=t211k
t211k=t211+t211n	t311k=t311+t311n
t221n=max(t211k,t121k)	t321n=max(t311k,t221k)
t221k=t221+t221n	t321k=t321+t321n
t231n=max(t221k,t131k)	t331n=max(t321k,t231k)
t231k=t231+t231n	t331k=t331+t331n
t241n=max(t231k,t141k)	t341n=max(t331k,t241k)
t241k=t241+t241n	t341k=t341+t341n
t251n=max(t241k,t151k)	t351n=max(t341k,t251k)
t251k=t251+t251n	t351k=t351+t351n
t261n=max(t251k,t161k)	t361n=max(t351k,t261k)
t261k=t261+t261n	t361k=t361+t361n

```

t411n=t311k
t411k=t411+t411n
t421n=max(t411k,t321k)
t421k=t421+t421n
t431n=max(t421k,t331k)
t431k=t431+t431n
t441n=max(t431k,t341k)
t441k=t441+t441n
t451n=max(t441k,t351k)
t451k=t451+t451n
t461n=max(t451k,t361k)
t461k=t461+t461n
РЕЗУЛЬТАТ=t461k

```

Аналогично были записаны следующие правила: в порядке возрастания левой половины технологического маршрута; в порядке убывания правой половины технологического маршрута; в порядке возрастания первой операции технологического маршрута; в порядке убывания последней операции технологического маршрута.

Когда все правила и соответствующие им циклы прохождения предметов труда по операциям последовательно записаны, осталось записать формулу по вычислению правила определяющего минимальную длительность цикла соответственно обсчитываемой матрице.

Дополнительно были записаны следующие формулы:
 $i = \min(\min(\min(t461k, t462k), \min(t463k, t464k)), t465k)$
 $p = \text{iiff}(i = t461k, 1, \text{iiff}(i = t462k, 2, \text{iiff}(i = t463k, 3, \text{iiff}(i = t464k, 4, \text{iiff}(i = t465k, 5, 0))))))$
 РЕЗУЛЬТАТ=p

Теперь посредством всех вышеуказанных функций, и с помощью программы, которой следует провести статистический эксперимент, можно обсчитать правила и выявить распределение случаев получения наименьшего времени длительности цикла соответственно разным правилам. Окно программы с результатом такого эксперимента приведено на рис. 1.

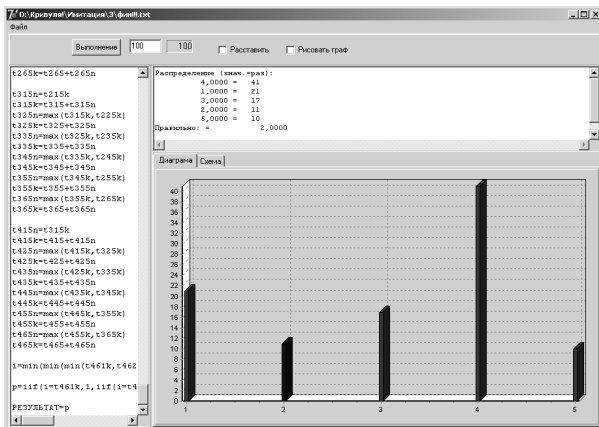


Рис. 1 Распределение случаев получения наименьшего времени длительности цикла соответственно разным правилам запуска

Статистический эксперимент показал, что запуск предметов труда в соответствии с четвертым правилом, длительность цикла оказывалась наименьшей большее число раз, чем в случае запуска по какому бы то ни было другому правилу. Отсюда следует рекомендация запуска предметов труда в порядке возрастания трудоемкости обработки по первой операции.

Таким образом, предприятие зная трудоемкости обработки предметов труда по операциям, пользуясь данной моделью, может эффективно организовывать свою работу. Дальнейшие исследования представляются в направлении поиска новых правил упорядочения.