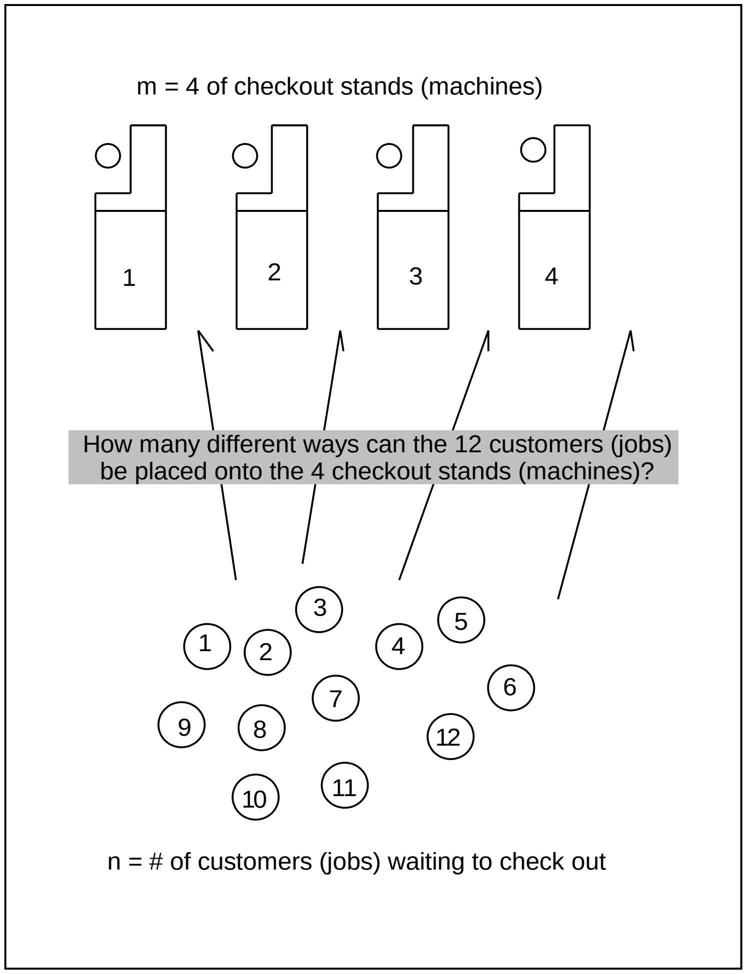
Appendix G

Consider the scenario below where n customers are waiting to be checked out on m number of checkout stands. How many different ways can the n customer sequenced on the m checkout stands?



Let's derive the $Perm_T$ formula below to find there number of permutations.

We know the answer:

$$m := 4 \quad n := 12$$

$$Perm_T := \frac{(m+n-1)!}{(m-1)!} = 2.179 \times 10^{11}$$

Job – Machine Arrangements Calculations

The following derivation provides a rule or formula to determine the number of arrangements (permutations) of n jobs on m parallel machines (i.e.; the number of ways n jobs can be sequenced on m machines where order **DOES** matter).

For example, consider two machines (m) in parallel required to process two jobs (A and B) without preemption. The number of all possible different sequences (arrangements) for machine loading are shown in Table G-1.

Table G-1 : Job Arrangements

Sequence	Machine 1	Machine 2
1	A, B	
2	B,A	
3	A	B
4	B	A
5		A, B
6		B, A

The problem is analyzed analytically by separating the problem into three parts determining:

1. Allocations of jobs on the machines
2. Partitioning of each job allocation
3. Permutations (distinct arrangements) for each partition

Substituting the formulas obtained in 2 and 3 into 1 permits development of a general formula for determining the number of distinct arrangements of n job sequenced on m parallel machines.

Part 1 – Job Allocation

Different combinations of jobs can be allocated on any machine. The possible allocations for n = 2 and m = 2 are shown in Table G-2.

Table G – 2 : Job Allocations

Allocation	Machine 1	Machine 2
1	2 jobs	
2	1 job	1 job
3		2 jobs

Determining the different number of job combinations is a combinatorial problem of ***unordered arrangement with repetition permitted***.

This problem does not permit decomposition into r-permutation, r-combination, or r-tuple rules since repetition is allowed. Our simple n = 2 and m = 2 problem can be explained by using a truncated domino game. Each domino is divided into two equal parts with each part containing either 1 or 2 spots, or blank (zero spots) [our game is identical to the real domino game except we use dominos possessing only 0, 1, or 2 spots unlike the real game which uses 0, 1, …, 6 spots]. Repetitions are allowed meaning each part of a domino may contain the same number of spots – double 2 for example.

We desire to know all possible ways of selecting 2 number from the set of 3 three numbers {0, 1, 2} *with repetition allowed*. The number of ways n objects can be selected from m objects with repetition is equivalent to the number of ways n objects can be selected from M + n – 1 objects without repetition.

The number of S(m,n) of unordered arrangements, with repetition (n-selections) is given by:

$$S(m,n) = C(m+n-1,n) = \begin{bmatrix} m+n-1 \\ n \end{bmatrix} = \frac{(m+n-1)!}{n!\cdot((m+n-1)-n)!}$$

Where, for our problem, n = the number of jobs and m = number of machines.

The proof S(m,n) = C(m + n –1,n) is shown on page 22 of [35].

Table G – 3, showing the number of job allocations for any n and any m, was generated using the S(m,n) formula described above. This table describes the different combinations of number of jobs that result for each n and m. It **does not** indicate the number of partitions or job sequencing permutations for each n and m. The permutation analysis will be determined in Section 3. High lighted allocations on Table G-3 were also confirmed empirically with the experiments shown in Table G-6 on the next page.

Table G-6 : Typical Job Allocations

n = 2, m = 2

Allocation	Machine 1	Machine 2
1	2 jobs	
2	1 job	1 job
3		2 jobs

n = 3, m = 2

Allocation	Machine 1	Machine 2
1	3 jobs	
2	2 jobs	1 job
3	1 job	2 jobs
4		3 jobs

n = 4, m = 2

Allocation	Machine 1	Machine 2
1	4 jobs	
2	3 jobs	1 job
3	2 jobs	2 jobs
4	1 job	3 jobs
5		4 jobs

n = 5, m = 2

Allocation	Machine 1	Machine 2
1	5 jobs	
2	4 jobs	1 job
3	3 jobs	2 jobs
4	2 jobs	3 jobs
5	1 job	4 jobs
6		5 jobs

The number of possible job allocations (combinations of numbers of jobs) form a skewed (diagonal) Pascal triangle indicating allocations are combinatorial in nature. The number of possible allocations for any n and m plays an important role in calculating the number of permutations (sequences) for any n and m as show in Part 3.

Table G - 3 : Number of Possible Allocations						
	Machines					
Jobs	1	2	3	4	5	6
1	1	2	3	4	5	6
2	1	3	6	10	15	21
3	1	4	10	20	35	56
4	1	5	15	35	70	126
5	1	6	21	56	126	252
6	1	7	28	84	210	462

Part 2 – Job Partitioning

While Table G-3 describes the number of all possible job allocations (combinations) of n jobs on m machines, it does not address the partitioning of the jobs on the machines. The **multinomial** equation provides all possible partitioning (**combinations - NOT permutations**) of n jobs onto m machines.

$$C\left(n, m_1, m_2 .. m_n\right) = \frac{n!}{m_1! \cdot m_2! .. m_n!}$$

Where $m_1 + m_2 + \ldots + m_n = n$ and n = number of jobs, $m_1!$ is the number of jobs processed on machine 1 and $m_2!$ the number of jobs processed on machine 2, etc.

Consider again Table G-2 showing all combinations of job allocations of 2 jobs on 2 machines:

Table G – 2 : Job Allocations

Allocation	Machine 1	Machine 2
1	2 jobs	
2	1 job	1 job
3		2 jobs

The total number of partitions for 2 jobs on 2 machines is just the **sum** of the partitions for all possible allocations thus:

$$Part_T = (Allocation\_1\_partions) + (Allocation\_2\_partions) + (Allocation\_3\_partitions)$$

or

$$Part_T = \frac{2!}{2!\cdot 0!} + \frac{2!}{1!\cdot 1!} + \frac{2!}{0!\cdot 2!} = 1 + 2 + 1 = 4$$

The 4 partitions of Table G-2's 3 job allocations are shown in Table G-4 where the 2 jobs are designated A and B.

Table G – 4 : Job Partitions

Allocation	Partition	Machine 1	Machine 2
1	1	A, B	
2	2	A	B
2	3	B	A
3	4		A, B

Note: partitions 2 and 3 are **not** permutations of allocation 2 but distinct partitions of the jobs A and B.

Part 3 – Job Permutations

Determining the partitions permits calculation of all possible permutations (sequences) of n jobs on m machines.

The number of permutations associated with any partition is just the product of the number of partitions and the permutations associated with that partition. Furthermore, each partition will have a fixed and known number of permutations. Below, Table G-5 visually depicts these permutations for n = 2 and m = 2.

Table G – 5 : Job Permuntations

Allocation	Partition	Permutation (sequence)	Machine 1	Machine 2
1	**1**	1	A, B	
1	**1**	2	B, A	
2	**2**	3	A	B
2	**3**	4	B	A
3	**4**	5		A, B
3	**4**	6		B, A
3 ALLOCATIONS	4 PARTITIONS	6 PERMUTATIONS		

For the case of n = 2 jobs and m = 2 machines, the number of permutations is clearly expressed by:

$Perm_T$ = (number of partitions in Allocation 1)(permutations of Allocation 1) + (number of partitions in Allocation 2)*(permutations of Allocation 2) + (number of partitions in Allocation 3)*(permutations of Allocation 3)+ etc.*

or

$$\mathrm{Perm}_T = \frac{n!}{\cancel{m_{1,1}!\cdot m_{1,2}!}}\cdot\left(\cancel{m_{1,1}!\cdot m_{1,2}!}\right) + \frac{n!}{\cancel{m_{2,1}!\cdot m_{2,2}!}}\cdot\left(\cancel{m_{2,1}!\cdot m_{2,2}!}\right) + \frac{n!}{\cancel{m_{3,1}!\cdot m_{3,2}!}}\cdot\left(\cancel{m_{3,1}!\cdot m_{3,2}!}\right)$$

$$\mathrm{Perm}_T = n! + n! + n! = 3\cdot n! = 6$$

(which is the number of permutations as shown above in Table G - 5)

Correspondingly, the number of distinct sequences (permutations) of n jobs processed by m machines is just (n!) * (number of possible job allocations from Part 3) where n is the number of jobs.

Thus, this number of distinct job sequences is given by:

$$\boxed{Perm_T = n!\cdot\frac{(m+n-1)!}{n!\cdot((m+n-1)-n)!} = n!\cdot\frac{(m+n-1)!}{n!\cdot((m-1))!} = \frac{(m+n-1)!}{(m-1)!} = \frac{\Gamma(m+n)}{\Gamma(m)}}$$

where:

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}\cdot e^{-x}\,dx$$

(the Gamma function)

Examples

Example # 1: Calculate the number of distinct sequences of processing n = 3 jobs on m = 3 machines.

$$Perm_T(3,3) = \frac{\Gamma(m+n)}{\Gamma(m)} = 60$$

On the following page, Table 7 shows empirically the distinct arrangements for n = 3 jobs scheduled on m = 3 machines to be 60.

Example # 2: Calculate the number of distinct sequences of processing n = 10 jobs on m = 3 machines.

$$Perm_T(10,3) = 23950080$$

Table G – 7: Empirical Solution (n = 3, m = 3)

	m1	m2	m3
1	1,2,3		
2	1,3,2		
3	2,3,1		
4	2,1,3		
5	3,1,2		
6	3,2,1		
7		1,2,3	
8		1,3,2	
9		2,3,1	
10		2,1,3	
11		3,1,2	
12		3,2,1	
13			1,2,3
14			1,3,2
15			2,3,1
16			2,1,3
17			3,1,2
18			3,2,1
19	1,2	3	
20	2,1	3	
21	2,3	1	
22	3,2	1	
23	3,1	2	
24	1,3	2	
25	3	1,2	
26	3	2,1	
27	1	2,3	
28	1	3,2	
29	2	3,1	
30	2	1,3	
31		1,2	3
32		2,1	3
33		2,3	1
34		3,2	1
35		3,1	2
36		1,3	2
37		3	1,2
38		3	2,1
39		1	2,3
40		1	3,2
41		2	3,1
42		2	1,3
43	1,2		3
44	2,1		3
45	2,3		1
46	3,2		1
47	3,1		2
48	1,3		2
49	3		1,2
50	3		2,1
51	1		2,3
52	1		3,2
53	2		3,1
54	2		1,3
55	1	2	3
56	1	3	2
57	2	3	1
58	2	1	3
59	3	1	2
60	3	2	1

60 possible permutations.

The number of distinct job sequences (as a function of n and m) is highly non-polynomial.

Using the $Perm_T$ formula developed above, Figure G-1 provides the number of distinct arrangements for **m = 3** while permitting the number of jobs to vary from n = 1 to n = 12.

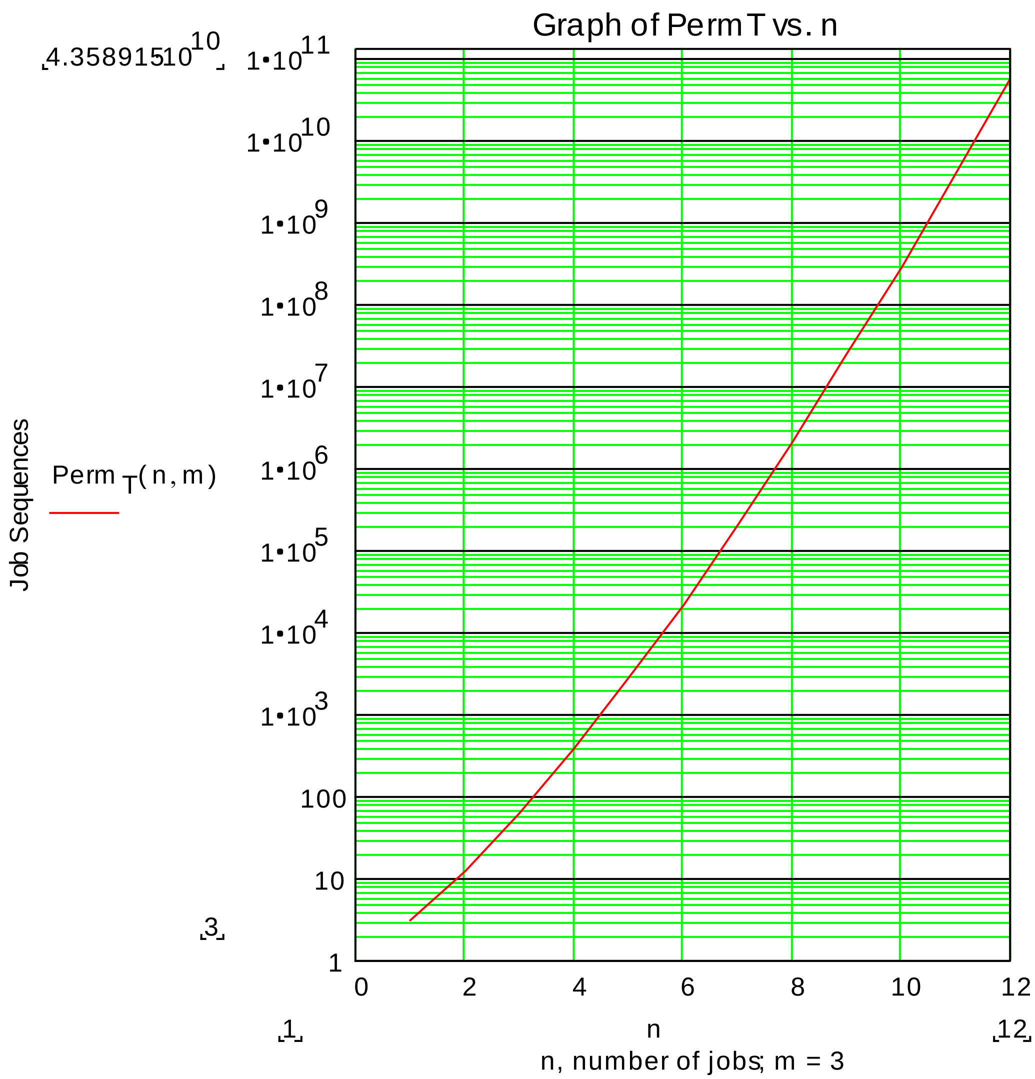


Figure G-1 : $Perm_T$ vs. n

Since the $Perm_T$ is a function of the two variables m and n, Figure G-2 depicts the 3 dimensional relationships of $Perm_T(m,n)$.

Growth in the number of possible job arrangements as a function of n jobs (from 1 to 6) and m machines (from 1 to 4).

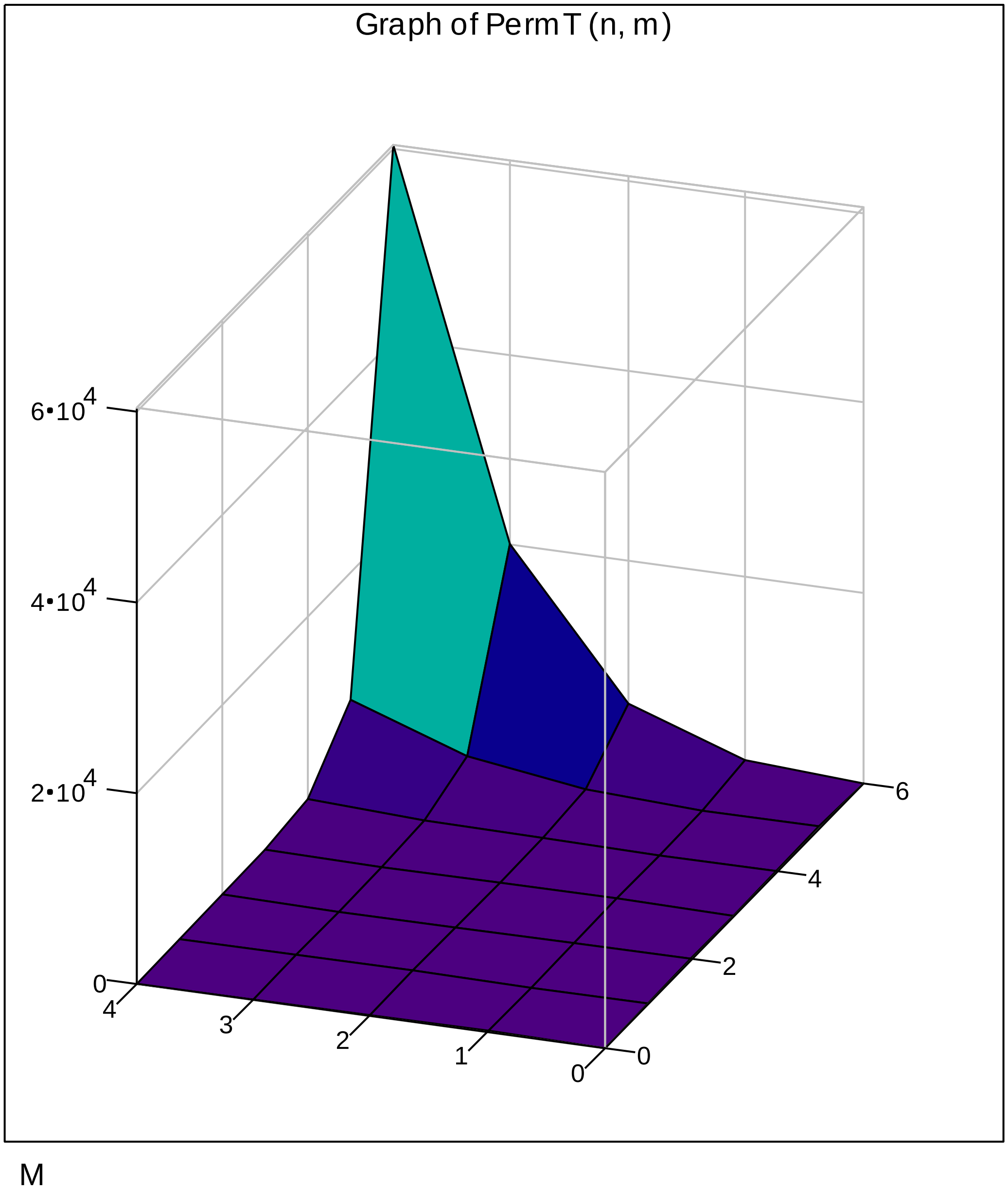


Figure G-2: $Perm_T$ vs. n and m