For come constant C in the naturals, and for a set of crews N, and a set of jobs J:

As N🡪 C, J 🡪 infinity, the difference between the maximum hours worked by crew K and the lower bound of job assignments JL goes to zero. For different data sets, this relationship is as follows:

|  |  |  |
| --- | --- | --- |
| Number of Crews (N) | Number of Jobs (J) | (K - JL ) / JL |
| 10 | 10 | 50% |
| 10 | 100 | 8.2% |
| 10 | 1000 | 0.9% |
| 10 | 10000 | 0.09% |
| 10 | 100000 | 0.008% |

In general, the trend seems to be the percentage of time the ‘overworked’ crew has directly approaches the ratio of N / J. I am not quite sure why this result holds. For the sample taken. It may have to do with how job times are created, as integers between (1, 8).

The runtime for the algorithm used to allocate jobs appears to be J\*N. The while loop runs until all jobs are allocated, and the inner loop checks for the smallest job allocation to any crew. Naively, this will be O(J\*N) time. This method was chosen simply because it was the simplest to create in the allotted time.