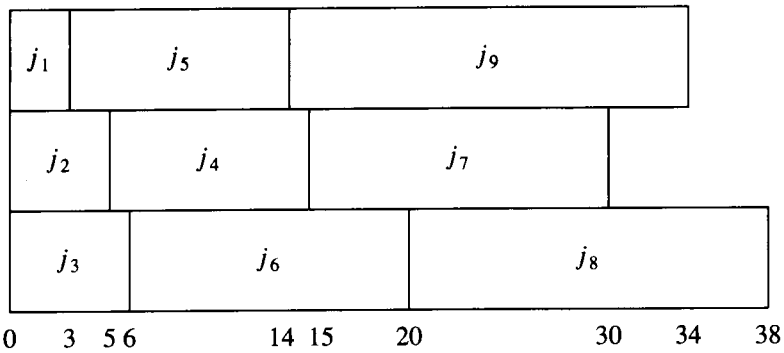

Figure 10.6 A second optimal solution for the multiprocessor case

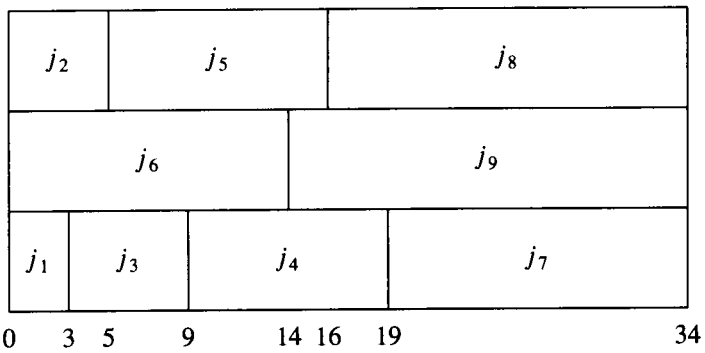

Figure 10.7 Minimizing the final completion time

owns all these jobs, then this is the preferable method of scheduling. Although these problems are very similar, this new problem turns out to be *NP*-complete; it is just another way of phrasing the knapsack or bin-packing problems, which we will encounter later in this section. Thus, minimizing the final completion time is apparently much harder than minimizing the mean completion time.

10.1.2. Huffman Codes

In this section, we consider a second application of greedy algorithms, known as *file compression*.

The normal ASCII character set consists of roughly 100 "printable" characters. In order to distinguish these characters, $\lceil \log 100 \rceil = 7$ bits are required. Seven bits allow the representation of 128 characters, so the ASCII character set adds some other "nonprintable" characters. An eighth bit is added as a parity check. The important point, however, is that if the size of the character set is C, then $\lceil \log C \rceil$ bits are needed in a standard encoding.

Suppose we have a file that contains only the characters *a*, *e*, *i*, *s*, *t*, plus blank spaces and *newlines*. Suppose further, that the file has ten *a*'s, fifteen *e*'s, twelve