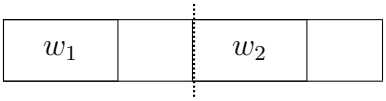
Chord Spacing

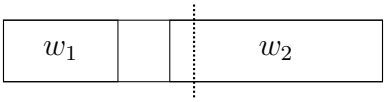
Consider the problem of placing chords in a bar so that they appear as close to their respective beats as possible[1]. Let's first solve the simpler problem with two chords in a bar. Given chord names of width w_1 and w_2 and width of the bar that will contain them w, where $w \geq w_1 + w_2$. We are to determine the widths of the spaces to be occupied by the two chords u_1 and u_2, where $u_1 \geq w_1$, $u_2 \geq w_2$, and $u_1 + u_2 = w$.

Clearly, the first chord will always be placed in the bar flushed left.

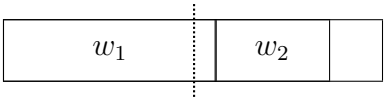
If $w_1 \leq w/2$, and $w_2 \leq w/2$, the second chord is best placed at mid-point. Therefore we set $u_1 = u_2 = w/2$.



If $w_1 \leq w/2$, and $w_2 > w/2$, the second chord is best placed flushed right in the bar. Therefore $u_1 = w - w_2$ and $u_2 = w_2$.



If $w_1 > w/2$, the second chord should be placed right next to it. Therefore $u_1 = w_1$ and $u_2 = w - w_1$.



For more than two chords, we recursively apply the above solution by letting w_1 and w_2 be widths of groups of chords. For example, to solve a problem with four chords, let w_1 and w_2 be the combined widths of the first two and last two chords, respectively. When the solution is found, the algorithm is applied recursively to place each of the two sets of chords. The formulation of the two subproblems requires resizing certain chords at the subproblem boundary. This is left as an exercise.