Reduction Algorithm

Given a graph $G(V, E)$, where V is the set of all of G's vertices and E all of G's edges, we wish to know if there exists k elements in any of its independent sets (NP-complete problem!). Let $n = |V|$ and $m = |E|$. We number the nodes as $b_1, b_2, \dots b_n$. We number the edges as $i_1, i_2, \dots i_m$. Each edge i_l is a distinct time interval. To reduce this independent set decision problem to Weighted Multiple Interval Scheduling (call it WMIS), Each node b_j is a job of value 1 and takes up time intervals represented by its incident edges. Call these jobs and their required intervals B. Input v to WMIS is k. The output answer from the imaginary WMIS code is the answer to our original independent set decision problem.

This reduction algorithm is polynomial – it is $O(VE)$, because we go to each node once and visit all of its edges. Converting the output to the desired output is constant time.

Proof of Correctness

This problem is NP-complete because it is both NP and NP-hard. It is NP because given a solution if the answer is yes, we can easily verify the correctness of the solution by adding up their values and seeing if any of them overlap.
This problem is NP-hard because the independent set problem is NP-complete and can be solved by reducing to WMIS. We prove the correctness of the reduction as follows.

1. If the answer to $IndSet(G, k)$ is yes, then the answer to $WMIS(B, k)$ is also yes
2. If the answer to $WMIS(B, k)$ is yes, then the answer to $IndSet(G, k)$ is also yes

1. If the answer to $IndSet(G, k)$ is yes, then there's at least k nodes $b_1, b_2, \dots b_k, \dots$ in an independent set of G. Their corresponding jobs could thus be scheduled without conflicts because they don't share any common intervals, which are represented by nodes being connected by edges.
2. If $WMIS(B, k)$ is yes, then that means at least k jobs $b_1, b_2, \dots b_k, \dots$ don't conflict with each other. This means their corresponding nodes in G form an independent set, as none of these jobs' intervals don't overlap, so none of the nodes are connected by an edge.