Performance of program analysis expressed as traversals over graphs like control flow graph (CFG) heavily depends on the order of nodes visited during the traversals: the traversal strategy, more so in case of BigCode analysis that performs analysis over a large collection of input graphs. While, there exists several choices for traversal strategy, like depth-first, post-order, reverse post-order, etc., there exists no technique to choose the most time-efficient strategy for traversals. In this paper, we propose hybrid technique that utilizes the static properties of the analysis, and the dynamic properties of the input graphs to select a most time-efficient strategy for each traversal on a graph. Our contributions are: a system for expressing program analysis as traversals, a set of static and dynamic properties that influence the traversal strategy selection, a set of static analyses to compute the properties, and a decision tree that checks the properties to select and optimize the most time-efficient traversal strategy. Our evaluation shows that the hybrid technique successfully selected the most time-efficient traversal strategy for 99.99%--100% of the time and using the selected traversal strategy, the running times of the analyses on BigCode in our evaluation were considerably reduced by 23%--79%. The overhead imposed by collecting additional information for our hybrid approach is less than 0.2% of the total running time for a large dataset and less than 0.01% for an ultra-large dataset.