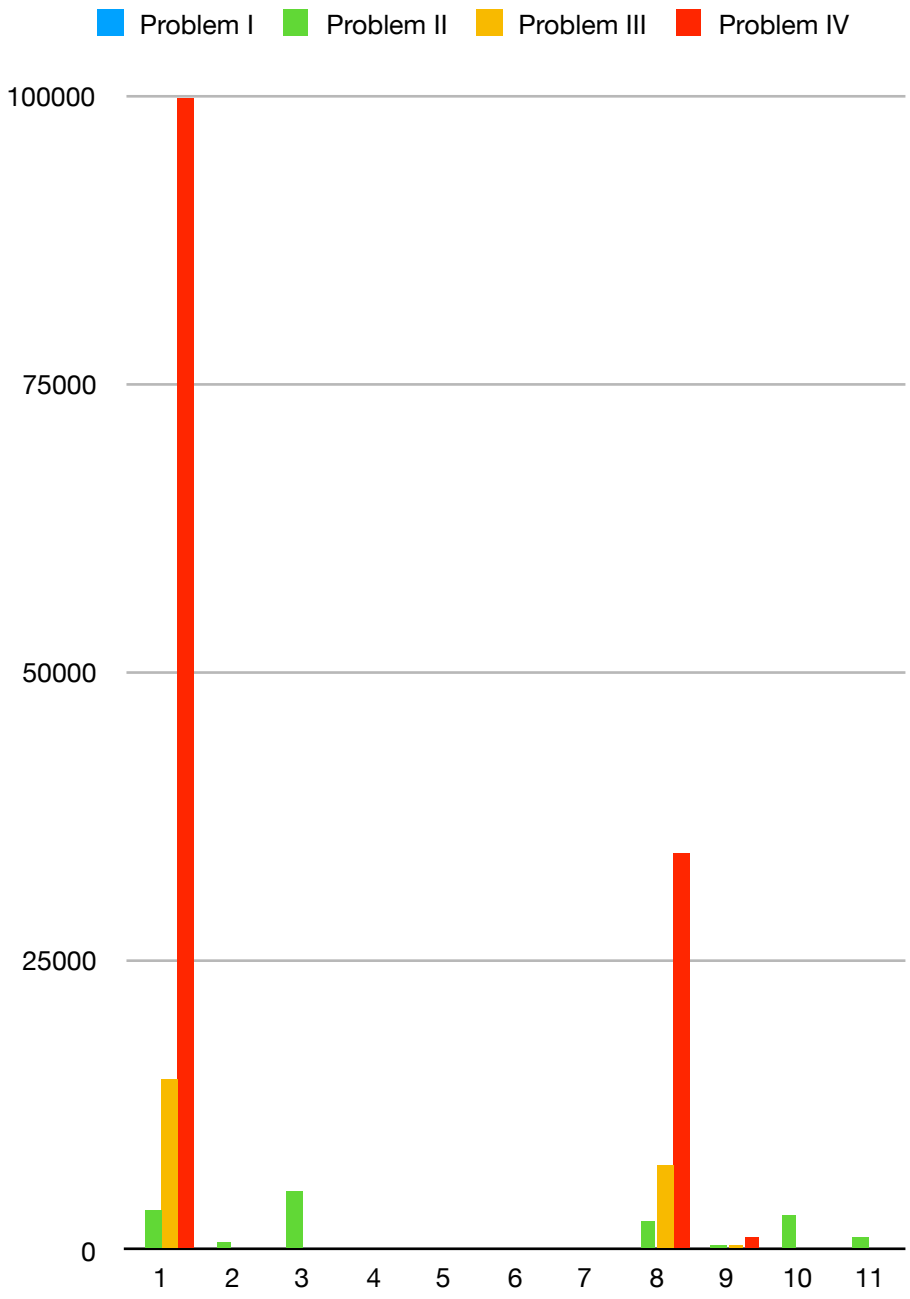
Build a forward planning agent

Nawfal Labrahmi

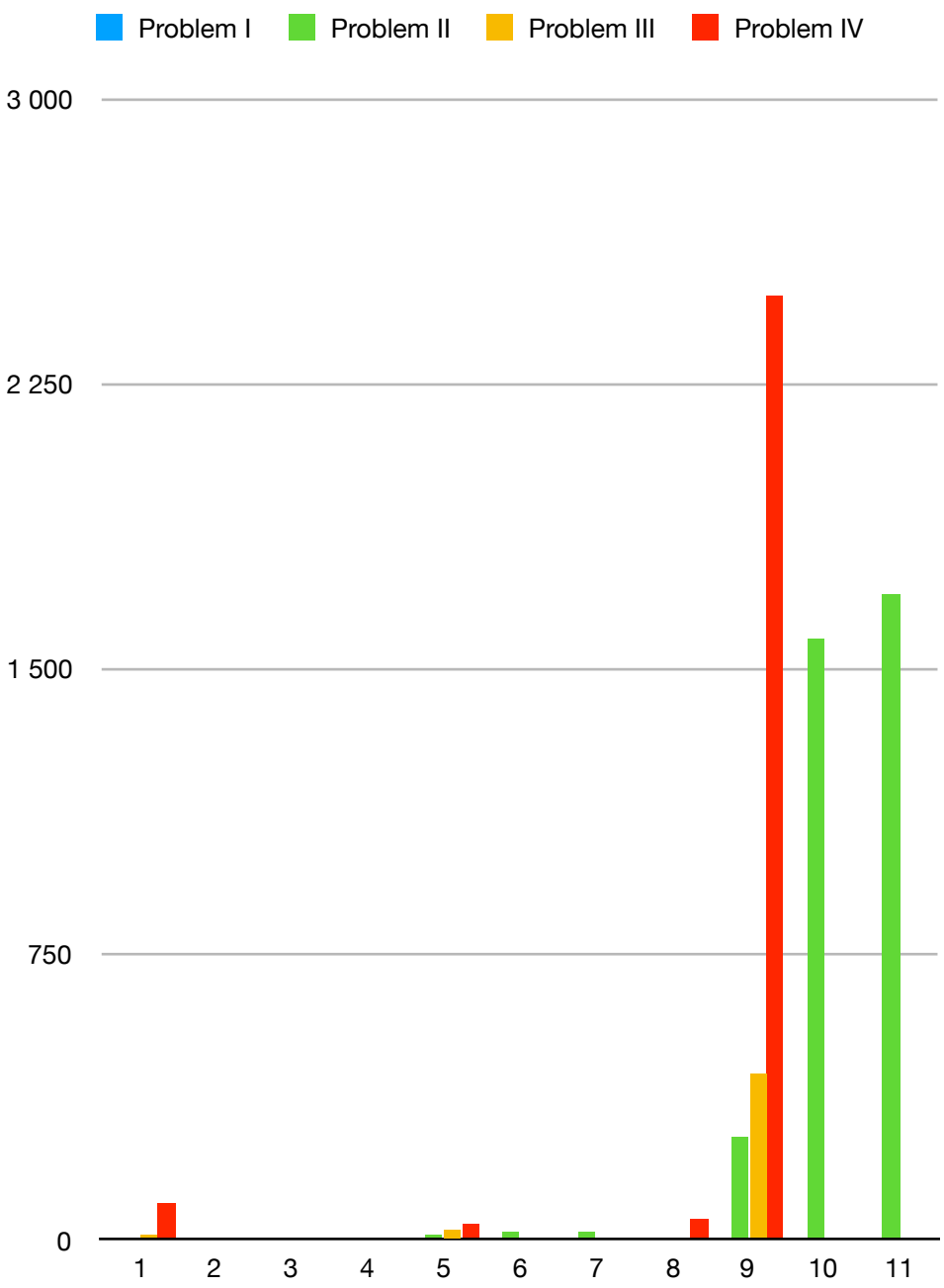
Analysis report

I. Search complexity as a function of domain size, search algorithm, and heuristic :



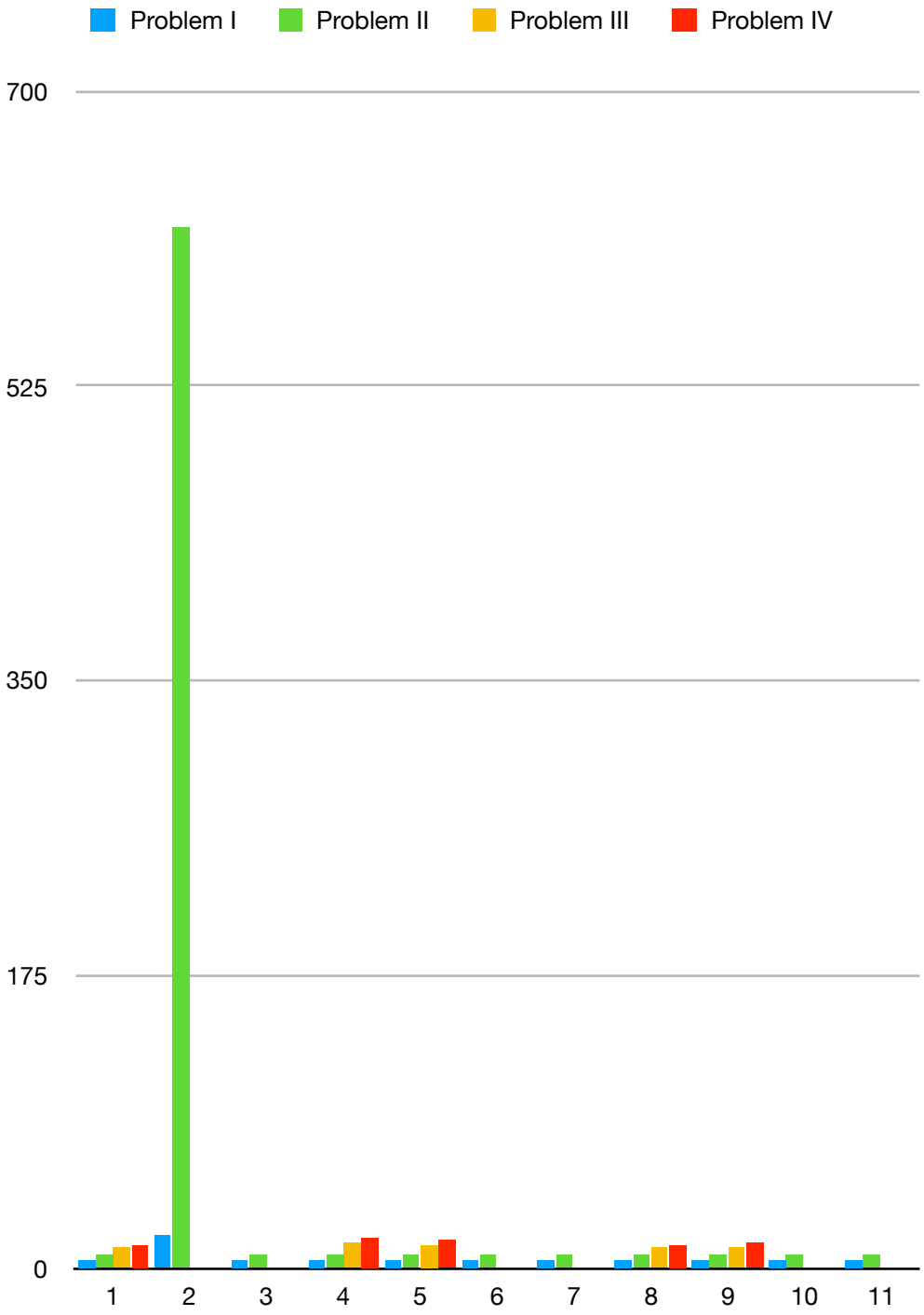
The graph shows common characteristics as well as divergence points between the different algorithms: if the domain size is small, all algorithms (and heuristics) have an insignificant search complexity, but as the domain size grows, some of them see their complexity explode (breadth-first search, A* with unmet_goals for eg.) when others (greedy best first for eg.) behave quite good even with bigger domain sizes.

II. Search time as a function of domain size, search algorithm, and heuristic:



Again, the data gathered shows great differences between the algorithms. The most obvious trend shown by the graph is that search time as far as A* algorithms (with the heuristics variants) literally explodes as the domain size grows. On the other hand, the greedy best first (with a special mention for unmet_goals heuristic) show very good results on growing domains.

III. Optimality of solution as a function of domain size, search algorithm, and heuristic:



Although there seem to be no big differences in the plan lengths (except for the obvious peak produced but depth-first algorithm), two key points are the following: the until now well behaved greedy best first seems to lose optimality with size, and for small domains, there are no major differences in optimality.

IV. Questions & Answers:

1. Which algorithm or algorithms would be most appropriate for planning in a very restricted domain (i.e., one that has only a few actions) and needs to operate in real time?

Operating in real-time means near to zero execution time, so the most appropriate (for very restricted domains) are greedy best first with unmet goals and depth first search.

2. Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)

The greedy best first search algorithms with their heuristic variants seem to behave quite good with large domains; they scale in a good fashion compared to other algorithms, although they may not deliver optimal plans.

3. Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?

If search complexity is not an issue, breadth first does the job perfectly well. If it is an issue, A* with unmet goals does better on the complexity side, while being able to deliver optimal plans.