**An explanation of the heuristic functions:**

For adversarial search, I have constructed the following heuristic:

A state’s utility is evaluated as , where is the difference between the number of people saved by agent one, and number of people saved by agent two; is the difference between the number of people carried by agent one and number of people carried by agent two; is the difference between number of people that agent one can reach sooner, and number of people that agent two can reach sooner.

Additionally, upon choosing between moves with equal utility, the playing agent will prefer a move that brings him to a shelter if it carries people. If there are multiple moves with equal utility that bring to a shelter, the agent chooses the one which is least time-consuming. If the agent does not carry people, it will prefer the least time-consuming move.

For both semi and fully-cooperative search, I have constructed the following heuristic:

A state’s utility is evaluated for both the playing agent and the other agent. The utility is evaluated as , where is the number of people saved by the agent; is the difference between the number of people that can be saved by the agent and the number of people that cannot be saved by the agent. For the purpose of this computation, the number of people that can be saved includes those the agent carries iff it can reach a shelter from the current state before deadline (taking and carry number into account), and those it can reach their location, and from there to a shelter in time (considering here just path lengths). The number of people that cannot be saved is the total number of people, minus those that have already been saved, minus those that we computed that can be saved.