Voting scenario: (i) **uncertainty in fire detection (mean voting yes no maybe, or probabilistic)**

(ii) **when fire is put out**

**Voting protocol: Median voter rule**

**Reason for this choice:**

The desired ~~qualities~~ **properties** of a voting protocol ~~in a domain are~~ **are** that ~~the protocol~~  **it** should be **Surjective** (i.e. every contestant should have an **equal** chance of winning), **Resolute** (i.e. there should always be a unique winner in the final round of voting), **Strategy Proof** (i.e. there should be no voter profile ~~where a~~ **such that** the voter prefers an outcome that is determined by its false preference) and **that** it should **not be a** **Dictatorship** (i.e. the winner should not be always identical to the first choice of some particular voter). ~~A voting~~ A desired protocol should also be **Weakly Pareto (i.e. if a particular candidate is not preferred by any voter then it will not be selected as a winner)** ~~(if there exists a candidate that no voter prefers then this candidate will not be selected as a winner)~~ and **Independent of Irrelevant Alternatives** (i.e. adding a new candidate will not make a candidate win that was previously not preferred).

**According to Gibbard-Satterthwaite Theorem, it is not possible to design a voting protocol that is Resolute, Surjective, Strategy Proof and not a Dictatorship. It states that a protocol can have at max any three of these desired properties but cannot have all of them.**

~~According to~~ **~~Gibbard-Satterthwaite Theorem~~**~~, if a protocol is Resolute, Surjective and Strategy Proof then it will be a dictatorship (i.e. there cannot be a protocol that is resolute, surjective and strategy proof but not dictatorship).~~

**Arrow’s Theorem states a similar limitation on these desired properties. It states that there cannot be a voting protocol that is Weakly Pareto, Independent of Irrelevant Alternatives and not a Dictatorship.**

~~According to~~ **~~Arrow’s Theorem~~**~~, if a protocol is weakly pareto and independent of irrelevant alternatives, it is a dictatorship (i.e. there cannot be a protocol that is weakly pareto, independent of irrelevant alternatives and not a dictatorship).~~

Since ~~deciding the presence of~~ **reporting** fire is the prime ~~decision for the domain~~ **functionality of the survey drones**, we desire a voting protocol that is Surjective, Resolute, Strategy proof, Weakly Pareto, Independent of Irrelevant alternatives and not a Dictatorship ~~(i.e. defy Gibbard-Satterthwaite and Arrow’s Theorem)~~. ~~This can~~ **~~only~~** ~~be achieved if we can remove the universal domain assumption by establishing a restriction on how agents can rank their preferences.~~

**To achieve this, we set up restrictions on the order of candidate preference that an agent can have while voting for the presence of fire.**

There are three candidates in this voting domain for an agent to decide a preference on: YesFire -- MaybeFire -- NoFire.

**The set of possible voting outcomes have a single peaked preference ordering on them as there exists a natural left-to-right arrangement on the order of candidate preference. So only the following candidate orderings are possible:**

**YesFire>>MaybeFire>>NoFire or NoFire>>MaybeFire>>YesFire or MaybeFire>>YesFire>>NoFire or MaybeFire>>YesFire>>NoFire**

~~There exists a single-peaked preference order on the set of possible voting outcomes as there exists a natural left to right arrangement on the .~~

~~While voting for presence of fire in the area, there exists a single-peaked preference order (YesFire-MaybeFire-NoFire) where there exists a natural arrangement from left to right. So only the following orderings are possible:~~

**~~YesFire>>MaybeFire>>NoFire~~** ~~or~~ **~~NoFire>>MaybeFire>>YesFire~~** ~~or~~ **~~MaybeFire>>YesFire>>NoFire~~** ~~or~~ **~~MaybeFire>>YesFire>>NoFire~~**

Median voter rule is an exception to both Gibbard-Satterthwaite and Arrow’s Theorem **as it removes the universal domain assumption by introducing restrictions on order of candidate preference**. ~~It is~~ **This voting protocol is** Surjective, Resolute, Strategy proof, Weakly Pareto, Independent of Irrelevant alternatives and not a Dictatorship. ~~We can use mean voter rule when agent is unsure of fire as there exists a natural right to left ordering in the preference in which agents can rank their choices.~~ This voting protocol is used for candidate voting under the following two scenarios.

**~~Scenario:~~**

a) **When a survey agent is unsure about the presence of fire (i.e. sensing heat levels below the alert threshold), it sends a voting signal to invite other survey drones from adjacent grids. The agents submit their preference order and appropriate action is taken after a winner is decided.**

~~An agent surveying block Z is unsure if there is fire in the block it is surveying. It sends a ‘MaybeFire’ signal in the communication network. The agents adjacent to block Z will leave their area and come to block Z, sense the area and vote their preference for fire.~~

b) When a firefighting drone has put out fire in a grid, it sends a voting signal to invite survey drones from adjacent grids. The winner of the voting determines the next actions.

**Voting in action:**

The agents have three candidates to order in voting preference: {YesFire, MaybeFire, NoFire}

The ways in which agents can choose their preference order is restricted as there exists a natural ordering on these candidate choices. Following are the possible ways of preference ordering:

**YesFire>>MaybeFire>>NoFire**

**NoFire>>MaybeFire>>YesFire**

**MaybeFire>>YesFire>>NoFire**

**MaybeFire>>YesFire>>NoFire**

**Outcomes:**

~~There are three outcomes of the voting: YesFire, NoFire, MaybeFire.~~

~~A fire fighting agent is called if the voting outcome is YesFire~~

**FIre is reported if the voting results in YesFire.**

~~Since the possible damage that could result from a false negative is very high, a firefighting agent is also called if the voting outcome is MaybeFire.~~

**Considering the high possibility of destruction due to a false negative, fire is reported even if the majority is unsure about the fire (MaybeFire wins).**

Firefighting agent is not called if the outcome of the voting is NoFire. This helps to reduce false positives and save resources that are used for putting off a fire.  
**Fire is not reported if NoFire is the outcome of the voting.**

**Draft 2**

The desired properties of a voting protocol are that it should be Surjective (i.e. every contestant should have an equal chance of winning), Resolute (i.e. there should always be a unique winner in the final round of voting), Strategy Proof (i.e. there should be no voter profile such that the voter prefers an outcome that is determined by its false preference) and that it should not be a Dictatorship (i.e. the winner should not be always identical to the first choice of some particular voter). A desired protocol should also be Weakly Pareto (i.e. if a particular candidate is not preferred by any voter then it will not be selected as a winner) and Independent of Irrelevant Alternatives (i.e. adding a new candidate will not make a candidate win that was previously not preferred).

According to Gibbard-Satterthwaite Theorem, it is not possible to design a voting protocol that is Resolute, Surjective, Strategy Proof and not a Dictatorship. It states that a protocol can have at max any three of these desired properties but cannot have all of them. Arrow’s Theorem states a similar restriction on these desired properties. It states that there cannot be a voting protocol that is Weakly Pareto, Independent of Irrelevant Alternatives and not a Dictatorship.

Since reporting fire is the prime functionality of the survey drones, we desire a voting protocol that is Surjective, Resolute, Strategy proof, Weakly Pareto, Independent of Irrelevant alternatives and not a Dictatorship. To achieve this, we set up restrictions on the order of candidate preference that an agent can have while voting for the presence of fire.

There are three candidates for this voting: YesFire - MaybeFire - NoFire.

The set of possible voting outcomes have a single peaked preference ordering on them as there exists a natural left-to-right arrangement on the order of candidate preference. So only the following candidate orderings are possible:

**YesFire>>MaybeFire>>NoFire or NoFire>>MaybeFire>>YesFire or MaybeFire>>YesFire>>NoFire or MaybeFire>>YesFire>>NoFire**

Median voter rule is an exception to both Gibbard-Satterthwaite and Arrow’s Theorem **as it removes the universal domain assumption by introducing restrictions on order of candidate preference**. **This voting protocol is** Surjective, Resolute, Strategy proof, Weakly Pareto, Independent of Irrelevant alternatives and not a Dictatorship. This voting protocol is used for candidate voting under the following two scenarios:

a) **When a survey agent is unsure about the presence of fire (i.e. sensing heat levels below the alert threshold), it sends a voting signal to invite other survey drones from adjacent grids. The agents submit their preference order and appropriate action is taken after a winner is decided.**

b) When a firefighting drone has put out fire in a grid, it sends a voting signal to invite survey drones from adjacent grids. The winner of the voting determines the next actions.

**Voting in action:**

Since there exists an ordering restriction on the predrences, agents can order the candidates in the following ways:

**YesFire>>MaybeFire>>NoFire**

**NoFire>>MaybeFire>>YesFire**

**MaybeFire>>YesFire>>NoFire**

**MaybeFire>>YesFire>>NoFire**

**Outcomes:**

**a) FIre is reported if the voting results in YesFire.**

**b) Considering the high possibility of destruction due to a false negative, fire is reported even if the majority is unsure about the fire (MaybeFire wins).**

c) **Fire is not reported if NoFire is the outcome of the voting.**

To be added in the temporal logic for sensing agents and firefighting agent:

sensing agent can send a signal for a fire\_voting

sensing agent has to participate in voting if it is in an adjacent block.

firefighting agent can start a fire\_voting

#can only start a fire\_voting

**Firefighting drone: [give(fire\_voting)]**

#can start and participate in a voting

**Sensing agent: ask(fire\_voting) [give(fire\_voting)]:**

**⌾ fact(fire\_voting) AND fact(adjacent\_grid)→ ◇ give(reach\_coordinates) AND give(vote\_preference)**

**Draft 3**

The desired properties of a voting protocol are:

* **Surjective:** Every contestant should have an equal chance of winning.
* **Resolute:** There should always be a unique winner in the final round of voting.
* **Strategy Proof:** There should be no voter profile such that the voter prefers an outcome that is determined by its false preference.
* **Not a Dictatorship:** The winner should never be always identical to the first choice of a particular voter.
* **Weakly Pareto:** If a particular candidate is not preferred by any voter then it will not be selected as a winner.
* **Independent of Irrelevant Alternatives:** Adding a new candidate will not make a candidate win that was previously not preferred.

According to Gibbard-Satterthwaite Theorem, it is not possible to design a voting protocol that is Resolute, Surjective, Strategy Proof and not a Dictatorship.

Arrow’s Theorem states that there cannot be a voting protocol that is Weakly Pareto, Independent of Irrelevant Alternatives and not a Dictatorship.

Since reporting fire is the prime functionality of the survey drones, we desire a voting protocol that is Surjective, Resolute, Strategy proof, Weakly Pareto, Independent of Irrelevant alternatives and not a Dictatorship. To achieve this, we set up restrictions on the order of candidate preference that an agent can have while voting for the presence of fire.

**Median voter rule**:

* Exception to Gibbard-Satterthwaite Theorem and Arrow’s Theorem
* It removes the universal domain assumption
* Introduces restrictions on order of candidate preference
* Domans should have a natural left to right ordering in candidate preferences

**Ω** = {**YesFire**, **MaybeFire**, **NoFire**} such that, **YesFire ≫ MaybeFire ≫ NoFire**

**Voting scenarios:**

a) When a survey agent is unsure about the presence of fire (i.e. sensing heat levels below the alert threshold), it sends a voting signal to invite other survey drones from adjacent grids. The agents submit their preference order and appropriate action is taken after a winner is decided.

b) When a firefighting drone has put out fire in a grid, it sends a voting signal to invite survey drones from adjacent grids. The winner of the voting determines the next actions.

**Voting in action:**

Since there exists an ordering restriction on the predrences, agents can order the candidates in the following ways:

**YesFire ≻ MaybeFire ≻ NoFire**

**NoFire ≻ MaybeFire ≻ YesFire**

**MaybeFire ≻ YesFire ≻ NoFire**

**MaybeFire ≻ NoFire ≻ YesFire**

**Possible Outcomes:**

YesFire wins: Fire alert is raised

NoFire wins: Fire alert is not raised

MaybeFire wins: Fire is reported as a precautionary measure

\\

a) FIre is reported if the voting results in YesFire.

b) Considering the high possibility of destruction due to a false negative, fire is reported even if the majority is unsure about the fire (MaybeFire wins).

c) Fire is not reported if NoFire is the outcome of the voting.