Auton Agent Multi-Agent Syst
DOI 10.1007/s10458-009-9118-5

Aggregating value ranges: preference elicitation and truthfulness

Joseph Farfel · Vincent Conitzer

Abstract We study the case where agents have preferences over ranges (intervals) of values, and we wish to elicit and aggregate these preferences. For example, consider a set of climatologist agents who are asked for their predictions for the increase in temperature between 2009 and 2100. Each climatologist submits a range, and from these ranges we must construct an aggregate range. What rule should we use for constructing the aggregate range? One issue in such settings is that an agent (climatologist) may misreport her range to make the aggregate range coincide more closely with her own (true) most-preferred range. We extend the theory of single-peaked preferences from points to ranges to obtain a rule (the median-of-ranges rule) that is strategy-proof under a condition on preferences. We then introduce and analyze a natural class of algorithms for approximately eliciting a median range from multiple agents. We also show sufficient conditions under which such an approximate elicitation algorithm still incentivizes agents to answer truthfully. Finally, we consider the possibility that ranges can be refined when the topic is more completely specified (for example, the increase in temperature on the North Pole given the failure of future climate pacts). We give a framework and algorithms for selectively specifying the topic further based on queries to agents.

1 Introduction

We consider settings in which a group of agents must decide on a range of values, based on the ranges that the individual agents consider "ideal." (In this paper, a *range* is an interval of real numbers.) For example, consider a group of climatologists debating by how much the earth's temperature will change over the next one hundred years. Suppose that each climatologist has privately-held minimum and maximum numbers of degrees by which she believes the temperature will change.[1] Also suppose that the climatologists have reached a point of fundamental disagreement (or distrust) and can no longer provide each other with

[1] These could also be the lower and upper bounds for what she believes is a (say) 99% confidence interval.

J. Farfel (✉) · V. Conitzer
Duke University, Durham, NC, USA
e-mail: joseph.farfel@gmail.com

Published online: 10 November 2009

any information that will make them update these beliefs. They would like to take a vote to produce a single aggregate temperature range (that, for example, they will then provide to the media).

Other applications abound. Lawmakers can vote on minimum and maximum jail sentences for certain illegal activities (here, the resulting minimum and maximum together constitute a range); program committee members can vote over a range for the quality of a paper (rather than a single number); etc. In all these applications, typically, the aggregated range is intended as a guideline that is of key importance for later decisions that the agents will make.

One may wonder why agents would have a most-preferred range rather than a single most-preferred number. There are at least the following three reasons. 1. *Indifference.* An agent may be indifferent among a range of values. 2. *Limited deliberation.* Determining one's ideal number can require significant effort in terms of deliberation, computation, or research; limited effort will result in a range of numbers (which will shrink as more effort is spent). For example, as a climatologist analyzes more ice cores, or runs more simulations, she will get a more precise idea of (smaller range for) the future increase in temperature. Similarly, a program committee member may read a paper more carefully and thereby get a more precise idea of (that is, shrink her range for) the quality of the paper. 3. *Underspecified topics.* An ideal climatologist may be able to give an exact number for the temperature increase if various specifics are provided to her—for example, that measurements will be taken on the North Pole, a particular climate pact will be enacted, *etc.* Without these specifics, however, she cannot possibly provide more than a range. Similarly, for the example of a lawmaker voting on the length of jail terms, the lawmaker may wish to allow some flexibility in the sentencing guidelines (that is, suggest a range of jail terms), to take into account the particulars and circumstances of the case.

An alternative model would be for each agent to have a probability distribution over point values. This may be reasonable if there is uncertainty over point values due to limited deliberation, or for some types of underspecification of the topic (one may have a probability distribution over which climate pacts will be enacted). However, it is not reasonable in the case of indifference, or for other types of underspecification (whether we are considering the North Pole or Antarctica). The range model can be applied more flexibly. More importantly, in practice, it is far more difficult to obtain a probability distribution from an agent than it is to obtain a range.

Our contributions in this paper will follow two main lines. First, we study what rule to use to produce an aggregate range from the individual ranges. A key issue here is that for naïve rules, such as taking a type of average of the ranges, an agent may have an incentive to lie about her range so as to make the aggregate range coincide more closely with her own most-preferred range. We propose a rule that (under a condition) is *strategy-proof*; that is, each agent is incentivized to report her true range. The condition generalizes the concept of *single-peaked preferences*, and the rule is the *median of ranges rule*.[2]

The bulk of the paper concerns the second line, where we study how to *elicit* agents' most-preferred ranges using simple queries (rather than making them reveal all of their information

[2] Of course, the importance of strategy-proofness depends on the particular application. For the climate example, one would hope that scientists would be interested not in making the aggregate range coincide with their own range, but rather in using the voting mechanism to discover the truth. Still, it is (unfortunately) likely that some parties involved in the determination of such an aggregate range would have a political objective, in which case using a strategy-proof rule seems desirable. But, perhaps the example of lawmakers voting on jail terms provides better motivation for using a strategy-proof rule, since here it does not seem that there is a "correct" answer to be discovered; rather, each lawmaker has her own preferences over ranges and it seems likely that each would want the aggregate range to coincide as closely as possible with her own true most-preferred range.

at once). We can approximate agents' most-preferred ranges to within a desired threshold using simple queries. Since the purpose of elicitation is to run the median of ranges rule, we study how to elicit an approximation of the median most-preferred range from a set of agents, without necessarily eliciting an approximation of every agent's most-preferred range. Eliciting ranges only approximately might introduce incentives for agents to answer queries falsely. We give sufficient conditions on the elicitation algorithm so that there is no such incentive. We also study the setting where there are more and less specified versions of a given topic, and we are interested in finding the agents' ranges for each version. Here, we study how to query the agents for the ranges of only a few versions in a way that lets us infer approximations of the ranges for other versions.

2 Review of single-peaked preferences

We first review some basic concepts from social choice (voting) theory, and specifically the concept of single-peaked preferences.

In a general voting setting, there is a set of agents (or *voters*) $\{1, \ldots, n\}$, who must select an alternative from a set O (which, in general, may be infinite). Each agent i has (privately held) preferences $\succ_i$ over O. For example, if $O = \{a, b, c\}$, agent i's preferences may be $b \succ_i a \succ_i c$, indicating that b is her most-preferred alternative and c her least preferred. Each agent reports preferences $\hat{\succ}_i$ (not necessarily her true preferences) over the alternatives. Then, a *voting rule* f takes the reported preferences (or *votes*) as input, and produces a winning alternative $f(\hat{\succ}_1, \ldots, \hat{\succ}_n)$. For example, the *plurality* rule chooses the alternative that is ranked first the most often. (In general, ties may occur, which must somehow be broken.) We note that in this case it suffices to ask each voter only for her most-preferred alternative. As another example, the *Copeland* rule is based on *pairwise elections*. In a pairwise election, we consider two alternatives a and b; the winner of the pairwise election is the alternative that is preferred by more voters (according to their reported preferences). Under the Copeland rule, an alternative receives two points for each pairwise election that it wins, one for each tie, and zero for each loss; the overall winner is the alternative with the most points.

Unfortunately, if voters can have *any* preferences over the alternatives, then there are various impossibility results that show that *no* voting rule obtains certain desirable properties. A voting rule is said to be *strategy-proof* if a voter never benefits from misreporting her preferences. That is, for any $\succ_1, \ldots, \succ_i, \ldots, \succ_n$, for any voter i, and any misreported preferences $\hat{\succ}_i$, either $f(\succ_1, \ldots, \succ_i, \ldots, \succ_n) = f(\succ_1, \ldots, \hat{\succ}_i, \ldots, \succ_n)$ or $f(\succ_1, \ldots, \succ_i, \ldots, \succ_n) \succ_i f(\succ_1, \ldots, \hat{\succ}_i, \ldots, \succ_n)$. The Gibbard–Satterthwaite theorem [11,16] shows that (when preferences are unrestricted and there are at least three alternatives) no voting rule is simultaneously *onto* (i.e., for every alternative, there exist votes that would make that alternative win), *nondictatorial* (i.e., the rule does not always choose the most-preferred alternative of the same voter), and strategy-proof. In the remainder of this paper, we will consider settings in which there are some restrictions on the voters' preferences, so that the Gibbard–Satterthwaite theorem does not apply.

In some settings, there is a natural order $<$ on the alternatives. For example, if the alternatives are political candidates, $<$ may represent which candidates are more left-wing. In the climate change assessment problem, the goal may be to produce a *single* number representing the projected temperature increase (rather than a range within which the increase is expected to lie). In this case, each alternative (projected temperature increase) is a real number, and $<$ is the natural order on the real numbers.

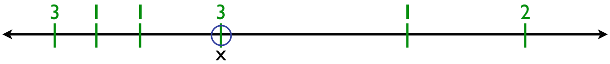

Fig. 1 An example with a continuum of alternatives, which receive 0, 1, 2, or 3 votes each. The median vote is *circled*

The order $<$ is of little relevance unless it is somehow related to the voters' preferences. One way in which the order may be related to the voters' preferences is that each voter prefers alternatives that are closer (according to $<$) to her most-preferred alternative. This intuition underlies the definition of *single-peaked* preferences [2].

Definition 2.1 Agent i's preferences $\succ_i$ are *single-peaked* with respect to $<$ if the following holds. If a is agent i's most-preferred alternative, then for any $b, c \in O$, we have 1. if $a < b < c$, then $b \succ_i c$; and 2. if $c < b < a$, then $b \succ_i c$.

We note that if $b < a < c$, then both $b \succ_i c$ and $c \succ_i b$ are consistent with single-peakedness. For example, according to the single-peakedness condition, a centrist voter must prefer a slightly left-wing candidate to a radically left-wing candidate, and must prefer a slightly right-wing candidate to a radically right-wing candidate; but she may or may not prefer a radically left-wing candidate to a slightly right-wing candidate.

Let us suppose that there is a *single* order $<$ over alternatives such that for *every* agent i, $\succ_i$ is single-peaked with respect to $<$. For simplicity, let us assume (throughout) that the number of voters n is odd.[3] Then, we can define a voting rule as follows. Let each voter i report only her most-preferred alternative $o(i)$. We will select the *median* alternative among these votes as the winner. That is, the winning alternative is the alternative a that receives at least one vote and for which $|\{i : o(i) \leq a\}| \geq (n+1)/2$ and $|\{i : a \leq o(i)\}| \geq (n+1)/2$. An example is given in Fig. 1. The median voting rule is well-known to have the following desirable properties (given the single-peakedness assumption):

- The winning alternative wins each of its pairwise elections (that is, it is the *Condorcet winner*).
- The voting rule is strategy-proof.

3 Extending single-peakedness to ranges

In the setting that we study in this paper, the alternatives that can be chosen are *ranges*, that is, intervals of real numbers. It seems reasonable to suppose that agents prefer ranges that are closer to their most-preferred range, but it is not immediately clear what the best definition of "closer" is. In this section, we generalize the theory of single-peaked preferences to the setting of ranges. (The classical theory presented in the previous section corresponds to the special case where all ranges consist of a single point.) We assume a finite, odd number of agents throughout.

There has already been some work on single-peaked preferences over multidimensional alternative spaces. Border and Jordan [3] generalize single-peaked preferences to Euclidian spaces of arbitrary dimension, and characterize the associated class of strategy-proof voting rules. Barberà, et al. [1] extend Border and Jordan's work, by examining situations where

[3] Without this assumption, the median is not well defined; this can be addressed by defining the median to be (say) the $(n/2)$th voter in the case of even n, but in order to keep notation simple we do not address this in this paper.

the space of alternatives is a compact subset of a Euclidean space, and proving that the class of strategy-proof voting rules in this setting consists of generalized median voter schemes (which satisfy another condition). In this section of our paper, we define single-peaked preferences over ranges, and show that a generalized median rule is strategy-proof. Since the set of alternative ranges can be thought of as a subset of $\mathbb{R}^2$, Barberà et al.'s result can in fact be used to prove this strategy-proofness result. We prove our result from first principles because it is more instructive and the proof is short (we do not need the full generality of Barberà et al.'s result).

3.1 Setup

The set of alternatives is a set of ranges, $O = \{[a, b] : a, b \in \mathbb{R}, a \leq b\}$. Each agent i has preferences $\succeq_i$ over O. We allow for these preferences to be *partial*; that is, there may be some pairs of ranges that are incomparable from the agent's perspective. (In more general settings, recent work has begun to extend social choice theory to allow for partial preferences [12–14], and partial preferences have also been studied from a more computational angle in social choice [19].) In general, each agent i reports an entire partial order $\hat{\succeq}_i$, and a voting rule f maps these reported preferences to a single range. (As we will see shortly, our voting rule does not actually require all this information from the agents.) We say that f is strategy-proof if for any $\succeq_1, \ldots, \succeq_i, \ldots, \succeq_n$, for any voter i, and any misreported preferences $\hat{\succeq}_i$, it is the case that $f(\succeq_1, \ldots, \succeq_i, \ldots, \succeq_n) \succeq_i f(\succeq_1, \ldots, \hat{\succeq}_i, \ldots, \succeq_n)$. This implies that the agent must be able to compare the range that results from her misreporting to the one that results from her telling the truth.

3.2 Definition of single-peakedness

If there is no restriction on the agents' preferences over alternatives (ranges), then the structure of the alternative space is irrelevant, and the Gibbard–Satterthwaite theorem applies (as well as many other impossibility results). So, we must assume that the agents' preferences are restricted in some natural way. We will do so by extending the notion of single-peaked preferences to ranges.

Definition 3.1 An agent has *single-peaked preferences over ranges* if she has a (most-preferred) range $P = [l^*, u^*]$ such that, for any ranges $X = [l, u]$ and $X' = [l', u']$ for which

- either $l' \leq l \leq l^*$ or $l^* \leq l \leq l'$; and
- either $u' \leq u \leq u^*$ or $u^* \leq u \leq u'$;

the agent weakly prefers X to X' ($X \succeq_i X'$).

An informal way of interpreting this definition is as follows: an agent's preferences over ranges are single-peaked if she has single-peaked preferences over the lower bound, and single-peaked preferences over the upper bound. It should be noted that, as in the case of "traditional" single-peaked preferences, there are many pairs of ranges for which any preference would be consistent with the definition.

We believe that the restriction imposed by our definition of single-peaked preferences over ranges is intuitively reasonable; it is straightforward to consider some examples to get some intuition (we will not provide detailed examples and discussion here, for the sake of space). On the other hand, there are certainly preferences that do not satisfy the condition.

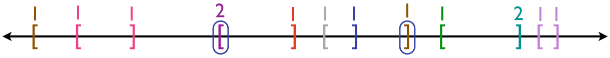

Fig. 2 *Labels* (1 and 2) indicate the number of value range bounds located at each point. The median value range bounds are *circled*. We note that the upper and lower bounds do not have to come from the same vote

For example, if an agent's primary concern is that the center of the aggregate range is close to the center of her own true range, this does not satisfy the condition. On the other hand, if the center of the range is the primary concern, then we effectively return to the standard setting with single-peaked preferences over points, and the ranges become somewhat inessential.

3.3 The median-of-ranges rule

Let us assume that all agents have single-peaked preferences over ranges. We now propose a voting rule that is strategy-proof given this restriction. Unsurprisingly, it is a generalization of the median rule.

Definition 3.2 The *median* of a set of ranges $\mathcal{S}$ is $M(\mathcal{S}) = [l_m(\mathcal{S}), u_m(\mathcal{S})]$, where $l_m(\mathcal{S})$ is the median of all lower bounds of ranges in $\mathcal{S}$, and $u_m(\mathcal{S})$ is the median of all upper bounds of ranges in $\mathcal{S}$.

Definition 3.3 The *median-of-ranges* rule takes each agent's most-preferred value range as input, and produces the median value range as output.

Figure 2 shows an example of the median range. We are now ready to show that the median-of-ranges rule is strategy-proof, given single-peaked preferences over ranges.

Theorem 3.1 *If all agents have single-peaked preferences over ranges, then the median-of-ranges rule is strategy-proof.*

Proof Let $P_i = [l^*, u^*]$ be any agent i's most-preferred value range, and let $\hat{P}_i = [\hat{l}, \hat{u}]$ be another range that i is considering reporting. Let $\hat{P}_{-i}$ be the ranges reported by other agents. Let $X = [l, u]$ be the median range when the agent reports P_i and the other agents report $\hat{P}_{-i}$. Let $X' = [l', u']$ be the median range when the agent reports $\hat{P}_i$ and the other agents report $\hat{P}_{-i}$.

We must show that agent i (weakly) prefers X to X'. Because i has single-peaked preferences, it suffices to show that either $l' \leq l \leq l^*$ or $l^* \leq l \leq l'$, and that either $u' \leq u \leq u^*$ or $u^* \leq u \leq u'$. By symmetry, it suffices to show this for the lower bounds. There are three possible cases:

1. If $l^* = l$, then either $l' \leq l = l^*$ or $l^* = l \leq l'$.
2. If $l < l^*$, then
   - If the agent reports $\hat{l}$ such that $l \leq \hat{l}$, then this does not change the median, so $l' = l < l^*$.
   - If the agent reports $\hat{l}$ such that $\hat{l} < l$, then this can only move the median to the left, so $l' \leq l < l^*$.
3. If $l^* < l$, then by reasoning analogous to case 2, it can be shown that $l^* < l \leq l'$.

In all three cases, either $l' \leq l \leq l^*$ or $l^* \leq l \leq l'$. □

The intuition for this result is simple: the median-of-ranges rule chooses the median of the submitted lower (upper) bounds as the output lower (upper) bound, every agent effectively has single-peaked preferences over the lower (upper) bound, and we know the median rule is strategy-proof when preferences are single-peaked.

One would hope that if all agents report ranges that are small, then the median range is also small. Certainly, this occurs in the limit when everyone submits a degenerate range (i.e., a single point), because in this case the output will also be a degenerate range (the median of these points). The following proposition shows that small reported ranges do indeed result in a small aggregate range.

Proposition 3.2 *If for each range $[l_i, u_i]$ being aggregated by the median-of-ranges rule, $u_i - l_i \leq k$, then for the resulting median range, $[l, u]$, we have $u - l \leq k$.*

Proof At least $(n + 1)/2$ ranges must have upper bounds $u_i \geq u$. Because each of these ranges has magnitude at most k, all of these ranges must have lower bounds $l_i \geq u - k$. Therefore, there are at least $(n + 1)/2$ lower bounds $l_i \geq u - k$, so it must be that $l \geq u - k$. □

4 Elicitation

So far, we have assumed that an agent can easily and precisely report her most-preferred value range. However, determining this precise range generally requires significant deliberation effort on the part of the agent. It would be preferable to have an *elicitation* algorithm that sequentially queries the agent for information about her most-preferred range, especially if the queries are natural and easy to answer. This has the advantages that 1) the agent receives some guidance in determining her range, and 2) few queries may already give us enough information about the agent's range for the purpose of (say) using the median rule, even if we do not yet know the agent's exact range—for example, if the agent's range is determined to be much further to the left than most other agents' ranges. As a result, elicitation can reduce the burden that participation in the mechanism places on an agent, thereby freeing up the agent to participate in other voting mechanisms, or to pursue other goals.

A significant body of research exists on eliciting agents' preferences (e.g., [4–7,17,18]). Elicitation techniques have been applied to combinatorial auctions (for an overview, see [15]) and also to voting settings [9,10]. Preference elicitation in single-peaked domains has been studied as well [8], but this work does not consider ranges and is primarily concerned with the case in which the order of alternatives on the line is not known.

4.1 Eliciting points

We will start by considering the special case where each agent i's range consists of a single point x_i. We will later generalize this to arbitrary ranges.

Definition 4.1 During the elicitation process, for each agent i, we maintain a *bounding range*, $B_i = [\underline{x_i}, \overline{x_i}\,]$, which bounds her privately held most-preferred point value, x_i.

We will consider elicitation using *bound queries*, which take the form: "Do you think the value should be greater than x?" For example, in the climate change example, a query might be: "Do you believe that the temperature will increase by more than 4 degrees?" This is arguably the most natural query in this type of domain.

Definition 4.2 A *bound query* is a query asking an agent i, with current bounding range $B_i = [\underline{x_i}, \overline{x_i}]$, if her privately held point value, x_i, is greater than some query point $x \in [\underline{x_i}, \overline{x_i}]$. If the agent answers "yes," set $\underline{x_i}$ to x, and if the agent answers "no," set $\overline{x_i}$ to x.

4.1.1 Eliciting a single agent's point

Mostly to get used to the notation, let us first consider the special case where we wish to elicit a single agent's point to within some threshold ϵ—that is, we terminate when $\overline{x_i} - \underline{x_i} \leq \epsilon$. A useful type of bound query is the *half-half query*, which queries an agent at the midpoint of her current bounding range ("Do you think the value should be greater than the midpoint of your current bounding range?") Such a query is guaranteed to cut the size of her bounding range in exactly half. Half-half queries give us the following simple result:

Proposition 4.1 *A single agent i's most-preferred point x_i can be elicited to within ϵ using at most* $\max\{\lceil \log_2 \frac{s}{\epsilon} \rceil, 0\}$ *half-half queries, where s is the size of the initial bounding range.*

4.1.2 Eliciting the median point

We now examine the problem of eliciting *multiple* agents' point values accurately enough to find the median point value to within some tolerance ϵ. We are not concerned with how accurately we have elicited any particular agent's point, as long as we have elicited enough information to find the median point to within ϵ.

Proposition 4.2 *Given bounding ranges $[\underline{x_i}, \overline{x_i}]$ for every agent i's most preferred point x_i, the lowest value that the median of the x_i can take is the median of the $\underline{x_i}$, and the highest value it can take is the median of the $\overline{x_i}$.*

This motivates the following definition:

Definition 4.3 The *median bounding range* is the range $M = [\underline{m}, \overline{m}]$, where $\underline{m}$ ($\overline{m}$) is the median of the $\underline{x_i}$ ($\overline{x_i}$) over all i.

The following property will be useful:

Proposition 4.3 *There is always an agent whose bounding range contains the entire median bounding range (that is, for some agent i, we have $\underline{x_i} \leq \underline{m}$ and $\overline{m} \leq \overline{x_i}$).*

Proof Let there be n agents in total (we note that n must be an odd number for the median to be defined). By the definition of median, there is a set $\mathcal{S}$ of at least $(n+1)/2$ agents where, for each agent s in the set, $\underline{x_s} \leq \underline{m}$. Also, there is the set $\mathcal{Q}$ consisting of no more than $(n-1)/2$ agents where, for each agent q in the set, $\overline{x_q} < \overline{m}$. Since $|\mathcal{Q}| < |\mathcal{S}|$, there is at least one agent $i \in \mathcal{S}$ for whom $\underline{x_i} \leq \underline{m}$ and $\overline{m} \leq \overline{x_i}$. □

To illustrate Proposition 4.3, we note that in Fig. 2, there is an agent whose range consists of the lowest lower bound and the median upper bound; this agent's range contains the entire median bounding range.

At any point in the elicitation process, queries to some agents may affect the median bounding range, whereas queries to other agents simply cannot.

Definition 4.4 A *relevant* agent is an agent i for which both $\underline{x_i} < \overline{m}$ and $\overline{x_i} > \underline{m}$ (that is, the agent's bounding range has an intersection with the median bounding range of nonzero measure).

Any query to an agent who is not relevant is wasted, as such a query cannot possibly move the position of either median bound. Such agents are irrelevant only insofar as querying them provides no new useful information; the location of an irrelevant agent's range (specifically, whether it is to the left or right of M) still affects the location of M. Another useful concept is that of agents who are relevant to shifting a median bound to a desired point x.

Definition 4.5 For a given point x inside M, an *x-relevant* agent is an agent i for whom $\underline{x_i} < x < \overline{x_i}$.

The notion of x-relevance allows us to prove the following lemma:

Lemma 4.4 *For a given point x inside M, let j be the initial number of x-relevant agents. Asking at most j bound queries at the point x to x-relevant agents will move the position of one median bound to x.*

Proof Each agent who is not x-relevant has her entire bounding range on one side of the point x (for such an agent i, either $\overline{x_i} \leq x$ or $x \leq \underline{x_i}$). When a bound query is asked to an x-relevant agent, her upper or lower bound moves to the point x, after which she is no longer x-relevant (by Definition 4.5).

Let $l(r)$ be the number of agents whose bounding range is entirely to the left (right) of x (so, $l + r$ is the total number of agents who are not x-relevant). Initially, l and r must each be less than $\frac{n+1}{2}$, since otherwise the point x would not be inside the median bounding range. Furthermore, l (r) can only increase when an agent's upper (lower) bound moves to x after a query (and neither l nor r can ever decrease). Since each x-relevant agent who is queried immediately loses x-relevancy, if we query all j initially x-relevant agents, all agents are no longer x-relevant, and we must at that point have $l + r = n$. It follows that after k queries, where $1 \leq k \leq j$, we have either $l = \frac{n+1}{2}$ with at least one upper bound located at x, or we have $r = \frac{n+1}{2}$ with at least one lower bound located at x. If $l = \frac{n+1}{2}$ $(r = \frac{n+1}{2})$, then there are $\frac{n+1}{2}$ upper (lower) bounds weakly to the left (right) of x, at least one of which is equal to x, so x must be the new median upper (lower) bound. □

Asking half-half queries is a natural way of eliciting a single agent's most-preferred point, since this halves the size of the agent's bounding range with each query. However, with multiple agents, we are interested in shrinking the median bounding range. Lemma 4.4 shows us how to ask queries to halve the size of this range. This motivates the following class of algorithms.

Definition 4.6 A *median halving algorithm* is any query algorithm that repeats these steps until convergence:

1. Set h equal to the current midpoint of M.
2. Ask bound queries at h to h-relevant agents until either $\underline{m}$ or $\overline{m}$ becomes h.

Let the initial size of the median bounding range be s. Then, for $s \geq \epsilon$, a median halving algorithm must halve the size of the median bounding range $\lceil \log_2 \frac{s}{\epsilon} \rceil$ times before convergence (analogously to Proposition 4.1). In the worst case, such a halving step requires n queries. However, if this happens, some agents must have become irrelevant, so that we do not need to query them in future steps.

Lemma 4.5 *Let x be some point inside the current median bounding range. If we ask $(n + 1)/2 + k$ (for some integer $k \geq 0$) bound queries at point x to x-relevant agents before moving one of the median bounds to x, then the number of relevant agents has decreased by at least k.*

Proof Each query asks an agent if her most-preferred point is greater or less than x. If the agent answers "greater," her lower bound is moved to x, while if she answers "less," her upper bound is moved to x. By Theorem 4.4, after enough queries one median bound will move to x. If the median upper (lower) bound did not move to x until $(n+1)/2+k$ queries were asked, it must be that at most $(n+1)/2$ of those agents queried answered "less" ("greater"), while at least k agents gave the opposite answer. This is because each agent is queried only once (she is no longer x-relevant after one query), and, by the definition of median, it is sufficient that $(n+1)/2$ agents give the same answer to the query to move the median upper (lower) bound to m. The (at least) k agents who answered "greater" ("less") are all now no longer relevant, because for each agent i in the set, we now have $\underline{x_i} \geq \overline{m}$ ($\overline{x_i} \leq \underline{m}$). □

Using Lemma 4.5, we can obtain a tight upper bound on the worst-case performance of any median halving algorithm.

Theorem 4.6 *A median halving algorithm takes at most*

$$\frac{n+1}{2}\max\left\{\left\lceil\log_2\frac{s}{\epsilon}\right\rceil, 0\right\}+\frac{n-1}{2}$$

queries to converge, where n is the number of agents, and s is the initial size of the median bounding range.

Proof Let the median bounding range be $M=[\underline{m},\overline{m}]$.

Consider one *halving step* in the query process to be as follows:

1. Set h equal to the current midpoint of M.
2. Ask bound queries at h to h-relevant agents until either $\underline{m}$ or $\overline{m}$ becomes h. □

One halving step results in (at least) halving the size of the median bounding range. Therefore, at most $\lceil\log_2\frac{s}{\epsilon}\rceil$ halving steps are needed to converge.

Intuitively, every halving step may require up to one query to every currently relevant agent (a halving step cannot require more than one query to a particular agent, because after she is queried at the point h, the point h is no longer strictly inside her range). The initial number of relevant agents might be n, and so one might imagine that every halving step queries n agents, in the worst case. However, Lemma 4.5 states that every time a halving step queries $\frac{n+1}{2}+k$ agents, for $k \geq 0$, at least k of those agents become irrelevant, and so will never be queried again. So, in the worst case, every halving step queries at least $\frac{n+1}{2}$ agents, and some halving steps query additional agents—but since each of these additional agents is made irrelevant after that query, that can happen at most once per agent throughout the entire process. This implies an additional $\frac{n-1}{2}$ queries (at most) in the entire process.

More precisely, let the number of queries needed to converge be $f(s,j)$, where s is the initial size of the median bounding range, and j is the initial number of relevant agents. We wish to show that

$$f(s,j)\leq b(s,j)$$

where $b(s,j)$ is the bounding function:

$$b(s,j)=\frac{n+1}{2}\max\left\{\left\lceil\log_2\frac{s}{\epsilon}\right\rceil,0\right\}+\max\left\{j-\frac{n+1}{2},0\right\}$$

(Note that showing this actually proves something slightly stronger than the result stated in the theorem. However, in the worst case, $j=n$, and the result stated in the theorem is the bound.)

We will show this inductively. As a base case, for any s where $0 \leq s \leq \epsilon$, for any j, zero queries are needed, while the bound is $\frac{n+1}{2} \max\{\lceil \log_2 \frac{s}{\epsilon} \rceil, 0\} + \max\{j - \frac{n+1}{2}, 0\} \geq 0$.

For the inductive step, assume that for some given s, we know that $f(\frac{s}{2}, j) \leq b(\frac{s}{2}, j)$, for any j. We now show that $f(s, j) \leq b(s, j)$. Because we already know that $f(s, j) \leq b(s, j)$ for $s \leq \epsilon$ by the base case, this will complete the proof. (We will assume $s > \epsilon$ in the below.)

- *Case 1:* The current halving step takes $q \leq \frac{n+1}{2}$ queries. Here, in the worst case, all j relevant agents remain relevant for the next halving step, because $b(\frac{s}{2}, j)$ is increasing in j. The number of queries to convergence is:

$$\begin{aligned} f(s, j) &\leq q + b\left(\frac{s}{2}, j\right) \\ &= q + \frac{n+1}{2}\left(\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil - 1\right) + \max\left\{j - \frac{n+1}{2}, 0\right\} \\ &\leq \frac{n+1}{2}\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil + \max\left\{j - \frac{n+1}{2}, 0\right\} = b(s, j) \end{aligned}$$

- *Case 2:* The current halving step takes $q = \frac{n+1}{2} + k$ queries, with $0 < k \leq \frac{n-1}{2}$. In this case, at most $j - k$ relevant agents remain relevant for the next halving step, by Lemma 4.5. So, the number of queries to convergence is:

$$\begin{aligned} f(s, j) &\leq \frac{n+1}{2} + k + b\left(\frac{s}{2}, j - k\right) \\ &= \frac{n+1}{2} + k + \frac{n+1}{2}\left(\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil - 1\right) + \max\left\{j - k - \frac{n+1}{2}, 0\right\} \end{aligned}$$

Since we are asking $\frac{n+1}{2} + k$ queries, and we only ask queries of relevant agents, it must be the case that $\frac{n+1}{2} + k \leq j$, and therefore we can evaluate the max statement to obtain:

$$\begin{aligned} f(s, j) &\leq \frac{n+1}{2} + k + \frac{n+1}{2}\left(\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil - 1\right) + j - k - \frac{n+1}{2} \\ &\leq \frac{n+1}{2}\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil + j - \frac{n+1}{2} = b(s, j) \end{aligned}$$

We have shown that $f(s, j) \leq b(s, j) = \frac{n+1}{2} \max\{\lceil \log_2 \frac{s}{\epsilon} \rceil, 0\} + \max\{j - \frac{n+1}{2}, 0\}$, for all $s \geq 0$, and for all integers $j \geq 0$. This implies the result of the theorem. □

To show that the bound given in Theorem 4.6 is tight, consider the following example for any ϵ, s with $\epsilon < s$ and any odd number n. Let the initial range be $[0, s]$. Let there be $(n+1)/2$ agents whose most-preferred point is $\epsilon/2$, and $(n-1)/2$ agents whose most-preferred point is $3s/4$. If the algorithm starts by querying the latter $(n-1)/2$ agents, then each of these agents will be queried once. The other $(n+1)/2$ agents will be queried $\lceil \log_2 \frac{s}{\epsilon} \rceil$ times, after which the median bounding range becomes a subset of $[0, \epsilon]$. So, the total number of queries in this example is

$$\frac{n+1}{2}\left\lceil \log_2 \frac{s}{\epsilon} \right\rceil + \frac{n-1}{2}$$

4.2 Eliciting ranges

We now return to the general setting where each agent i has a most-preferred value range $P_i = [l_i, u_i]$.

Definition 4.7 For each agent i, we maintain a *lower bounding range* $L_i = [\underline{l_i}, \overline{l_i}]$ and an *upper bounding range* $U_i = [\underline{u_i}, \overline{u_i}]$.

We will elicit ranges using a generalized form of the bound query.

Definition 4.8 A *range bounding query* asks an agent i if she believes the value should be greater than some point x. There are three possible answers:

1. "yes:" The agent's entire range P_i is greater than x. In this case, we set $\underline{l_i}$ to x, and if $\underline{u_i} < x$, we also set $\underline{u_i}$ to x.
2. "no:" The agent's entire range P_i is smaller than x. In this case, we set $\overline{u_i}$ to x, and if $x < \overline{l_i}$, we also set $\overline{l_i}$ to x.
3. "maybe:" The point x is inside the agent's range P_i. If $x < \overline{l_i}$, then we set $\overline{l_i}$ to x; and if $\underline{u_i} < x$, then we set $\underline{u_i}$ to x.

4.2.1 Eliciting a single agent's range

Once again, we start with the case of a single agent. We propose a three-stage algorithm to elicit a single agent's range using range bounding queries. This algorithm finds both the lower bound and the upper bound of the agent's range P_i to within ϵ.

The algorithm consists of three binary searches: one to find a point inside the agent's range, one to find the lower bound, and one to find the upper bound. Each time the algorithm asks agent i a range bounding query, it is implied that the agent's bounds are updated as specified in the definition of a range bounding query.

Let L be an initial lower bound on agent i's range, and let U be an initial upper bound —for instance, in the climate change example, we could have $L = -100$ and $U = 100$. We initialize $\underline{l_i} = \underline{u_i} = L$, and $\overline{l_i} = \overline{u_i} = U$.

Stage 1: Ask a range bounding query at the point $x = (\overline{u_i} + \underline{l_i})/2$. Repeat until either a "maybe" answer is given (in which case the algorithm proceeds to Stage 2), or $\overline{u_i} - \underline{l_i} \leq \epsilon$ (in which case the algorithm terminates).
Stage 2: Ask a range bounding query at the point $x = (\overline{l_i} + \underline{l_i})/2$. Repeat until $\overline{l_i} - \underline{l_i} \leq \epsilon$.
Stage 3: Ask a range bounding query at the point $x = (\overline{u_i} + \underline{u_i})/2$. Repeat until $\overline{u_i} - \underline{u_i} \leq \epsilon$.

Figure 3 shows an example run of the algorithm.

Theorem 4.7 *Assuming $U - L \geq \epsilon$, the above algorithm requires at most $2\lceil \log_2 \frac{U-L}{\epsilon} \rceil - k$ queries, where k is the number of queries in stage* 1 $(1 \leq k \leq \lceil \log_2 \frac{U-L}{\epsilon} \rceil)$.

Proof Each query in Stage 1 for which the answer is not "maybe" halves $\overline{u_i} - \underline{l_i}$, so there can be at most $\lceil \log_2 \frac{U-L}{\epsilon} \rceil$ queries in this stage. Now suppose that we reach Stage 2 after asking k queries in Stage 1. Then, at the beginning of Stage 2, $\overline{l_i} - \underline{l_i}$ is at most $(1/2)^k (U - L)$. Each query in Stage 2 halves $\overline{l_i} - \underline{l_i}$, so there can be at most $\lceil \log_2 \frac{U-L}{\epsilon} \rceil - k$ queries in this stage. By symmetry, Stage 3 also requires at most $\lceil \log_2 \frac{U-L}{\epsilon} \rceil - k$ queries. It follows that the total number of queries is at most $2\lceil \log_2 \frac{U-L}{\epsilon} \rceil - k$. The best case is when $k = \lceil \log_2 \frac{U-L}{\epsilon} \rceil$; that is, the algorithm terminates in Stage 1. The worst case is when $k = 1$; that is, the first answer is "maybe." □

In a sense, the first stage is twice as efficient as the second and third stages, because a query in the first stage will halve both the lower and upper bounding ranges.

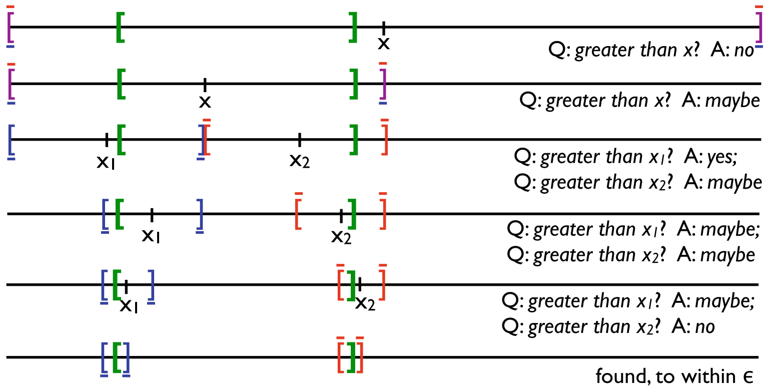
x
Q: greater than x? A: no
x
Q: greater than x? A: maybe
x1
x2
Q: greater than x1? A: yes;
Q: greater than x2? A: maybe
x1
x2
Q: greater than x1? A: maybe;
Q: greater than x2? A: maybe
x1
x2
Q: greater than x1? A: maybe;
Q: greater than x2? A: no
found, to within ϵ

Fig. 3 An example run of the elicitation algorithm. *Bold brackets* indicate the true range, while *other brackets* indicate the algorithm's current bounds. To save space, Stages 2 and 3 are shown running in parallel

4.2.2 Eliciting the median range

We are finally ready to consider the fully general case, in which there are multiple agents with ranges and we wish to elicit the median range to within ϵ.

Definition 4.9 The *lower median bounding range* is the range $M_l = [\underline{m_l}, \overline{m_l}]$, where $\underline{m_l}$ is the median of all $\underline{l_i}$, and $\overline{m_l}$ is the median of all $\overline{l_i}$. The *upper median bounding range* $M_u = [\underline{m_u}, \overline{m_u}]$ is defined similarly.

In this setting, it turns out that we can use the median halving algorithm to find the median lower bound to within ϵ, and subsequently we can use the median halving algorithm to find the median upper bound to within ϵ. To use a median halving algorithm to find (say) the median lower bound, we need to be able to ask an agent a query of the form "Is your lower bound l_i greater than x?" We can simulate this query using a range bounding query at x: if the agent answers "yes" to the range bounding query, this means that l_i is greater than x; if she answers either "no" or "maybe" to the range bounding query, this means that l_i is smaller than x. The case of finding the median upper bound is similar.

Definition 4.10 A *median range halving algorithm* is any query algorithm that has this structure:

1. Find the median lower bound to within ϵ using a median halving algorithm, simulating queries about agents' lower bounds using range bounding queries. If the answers to the range bounding queries give information about the agents' upper bounds, update these as well.
2. Find the median upper bound to within ϵ using a median halving algorithm, simulating queries about agents' upper bounds using range bounding queries.

Theorem 4.8 *Assuming $s \geq \epsilon$, a median range halving algorithm takes at most $(n+1)\lceil \log_2 \frac{s}{\epsilon} \rceil + n - 1$ queries to converge, where n is the number of agents, and s is an upper bound on the initial size of each of the two median bounding ranges.*

Proof In a median range halving algorithm, we find the median lower bound to within ϵ (Step 1), and then find the median upper bound to within ϵ (Step 2). Step 1 is a median

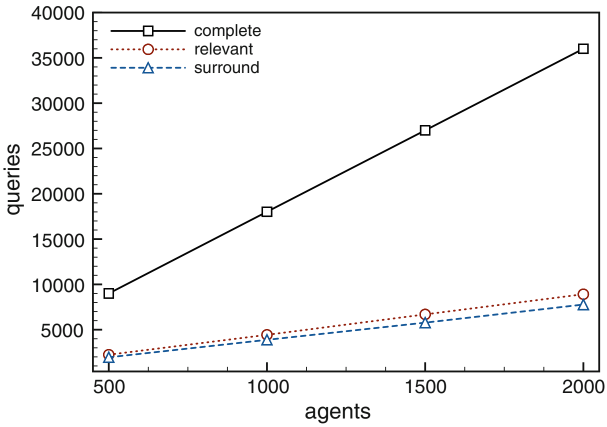

Fig. 4 The number of queries needed by different median range elicitation algorithms. In each experiment, simulated agents were queried until the median range was found to within $\epsilon = 0.001$. Each agent's most-preferred range was created from a random draw (from a uniform distribution) of two numbers between 0 and 1. Each data point is averaged over 100 runs. "Complete" and "relevant" use only half-half queries. " Complete" queries every agent to within ϵ. " Relevant" queries (only) relevant agents in turn, until convergence. "Surround" is a median range halving algorithm; each agent it chooses to query is a random agent whose range surrounds the median range (by Proposition 4.3 such an agent always exists). While a huge improvement is gained simply by not asking irrelevant agents, it is worth noting that "surround" consistently asks about 13% fewer queries than the more naïve "relevant."

halving algorithm with an initial median bounding range size of at most s, so by Theorem 4.6 it requires a maximum of $\frac{n+1}{2}\lceil\log_2 \frac{s}{\epsilon}\rceil + \frac{n-1}{2}$ queries. The same analysis applies to Step 2.[4] □

Figure 4 shows the results of a simple experiment comparing the performance of different median range elicitation algorithms.

5 Elicitation incentives

Since the median rule (and the median of ranges rule) is strategy-proof for agents with single-peaked preferences over points (or ranges), for any elicitation algorithm that results in these exact points (ranges), answering queries truthfully is an ex-post equilibrium.[5] However, if the algorithm only finds an approximation to the median, this may introduce incentives for

[4] We note that Step 1 may reduce the number of queries needed by Step 2, because range bounding queries can move two bounds in one query (a bound on an agent's upper bound and a bound on an agent's lower bound). This could cause the size of the median upper bounding range to decrease before the beginning of Step 2, but this can only help.

[5] A profile of strategies for the agents is an ex-post equilibrium if, regardless of the agents' preferences, it is optimal for every agent to follow her strategy, given that the other agents do so as well. This is weaker than saying that these strategies are dominant strategies, because a player's ex-post equilibrium strategy is in general not optimal if the other players are not following the ex-post equilibrium. It is well known that if we take a direct-revelation mechanism that is strategy-proof (dominant-strategy truthful), and then make the modification that we use an elicitation algorithm to obtain the preferences for this mechanism, then answering queries truthfully is an ex-post equilibrium. In general, answering queries truthfully will not be a dominant strategy for an agent, because in principle other agents could have strategies that condition on the first agent's answers (which cannot happen when all agents report their preferences simultaneously).

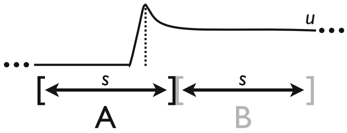

Fig. 5 An example agent's single-peaked utility function over point values. In this example, the agent does not have SPPESNOR if the utility the agent receives for a mechanism outputting a range M is equal to the integral of the agent's utility function over M divided by $|M|$. In this case, the agent prefers the range B to the range A, even though the agent's most-preferred point is in range A, and both ranges have the same size, s

answering queries falsely. In this section we will give sufficient conditions on the algorithm so that answering truthfully is an ex-post equilibrium in spite of the approximation.

5.1 Median point elicitation incentives

Again, we start with the simpler setting where each agent i's range consists of a single value x_i. We are concerned with query algorithms (such as the median halving algorithm) that approximate the median point by keeping a bounding range on it during the query process. The output of such an algorithm is the final median bounding range. We must make some assumption about agents' preferences over these outcome ranges, if we are to prove that these querying algorithms incentivize truthful answering. The following assumption seems quite minimal.

Definition 5.1 Agent i, whose most-preferred point is x_i, has *single-peaked preferences over equally-sized non-overlapping ranges (SPPESNOR)* if, for any ranges X and X' where

1. $|X| = |X'|$ (they are the same size);
2. $X \cap X'$ contains at most one element (they do not overlap, or they overlap only at an endpoint); and
3. either
   - X lies between X' and x_i, or
   - X contains x_i

agent i weakly prefers X to X' ($X \succeq_i X'$).

If agents have single-peaked preferences over output points, then agents have SPPESNOR in any scenario where the utility an agent receives for some range M being outputted by a mechanism is equal to the maximum of the agent's utility function over the points in M. However, one might imagine other scenarios where agents have single-peaked preferences over points, and yet they do not have SPPESNOR. If the utility an agent receives for a mechanism outputting a range M is equal to the integral of the agent's utility function over M divided by $|M|$, then the agent only has SPPESNOR if the utility function is symmetric.

Figure 5 shows a counterexample for the case where it is not symmetric. Also, Fig. 6 shows an example of why SPPESNOR specifies that the two ranges being compared are the same size.

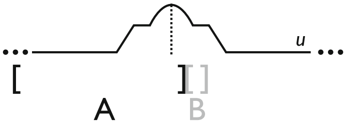

Fig. 6 An illustration of why SPPESNOR specifies that the ranges being compared are the same size. If the utility the agent receives for a mechanism outputting a range M is equal to the integral of the agent's utility function over M divided by $|M|$, then this agent prefers the range B to the range A, even though the agent's most-preferred point is in range A, and the agent's utility function is symmetric

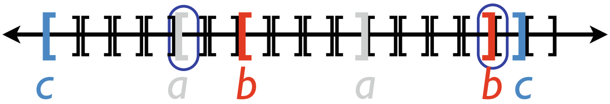

Fig. 7 An illustration of a partitioning query algorithm in action. The partition into alternative ranges is shown, along with three agents' current bounding ranges, and the current median bounding range (*circled*)

5.1.1 Partitioning the space of possible points

We will consider algorithms that partition the space of values into a set of (small) equally-sized alternative ranges,[6] and produce one of these ranges as the output. We will show later that the algorithms that we have considered so far do in fact have this property.

Definition 5.2 A *partitioning query algorithm* is any query algorithm where

1. The space of possible values is partitioned into a set $\mathcal{S}$ of alternative ranges, each with the same size $z \in (\epsilon/2, \epsilon]$.
2. Each query to an agent is a bound query at some point x which is at the boundary of one or two alternative ranges.
3. Querying continues until the median bounding range consists of a single alternative range in $\mathcal{S}$.

Figure 7 illustrates the idea of a partitioning query algorithm.

Theorem 5.1 *If all agents have SPPESNOR, then for any partitioning query algorithm, answering truthfully is an ex-post equilibrium.*

Proof Let there be s ranges in the set $\mathcal{S}$ that partitions the set of possible points. The s ranges in $\mathcal{S}$ can be thought of as s alternatives to be voted on by the agents.

We first show that agents have single-peaked preferences over these alternatives. Consider any arbitrary agent i, with most-preferred point x_i. Let $A = [\underline{a}, \overline{a}]$ and $B = [\underline{b}, \overline{b}]$ be two ranges from $\mathcal{S}$ on the same side of x_i, and let A be closer to x_i than B is (where neither contains x_i). Since no alternative ranges are overlapping (with the exception of boundary points), the agent must weakly prefer the alternative A to the alternative B, since she has SPPESNOR. Furthermore, SPPESNOR implies that the agent weakly prefers the alternative range P containing her most-preferred point x_i to any other alternative range B. (If x_i is a boundary point between two alternatives, P and P', each of the two alternatives is weakly preferred to any other, including each other.) So, the agent has a most-preferred alternative (P), and for any

[6] We consider these ranges to include both of their endpoints, so that the term "partition" is slightly inaccurate because boundary points will in general be part of two ranges.

two alternatives on the same side of P, she weakly prefers the alternative closer to P. The agent therefore has single-peaked preferences over the alternative ranges in $\mathcal{S}$.

When we elicit using bound queries located at boundary points of ranges in $\mathcal{S}$, this is equivalent to eliciting over the alternatives in $\mathcal{S}$. A bound query at a point x asks an agent if her most preferred alternative is to the left of x or to the right. Since each bound query is on a boundary between two alternatives in $\mathcal{S}$, and no alternatives overlap (except at the boundary points), each agent's bounding range will always contain a subset of whole alternative ranges from $\mathcal{S}$. Further bound queries to this agent must be at a point that is both within her range and is one of the boundary points of a range in $\mathcal{S}$, so each query will shrink the agent's bounding range so it contains at least one fewer alternative than before. Once an agent's bounding range contains only one alternative range, she cannot be queried further – she has selected this alternative as her most preferred. Proposition 3.2 guarantees that, if we query each agent to a range with size no larger than ϵ, then the median range has size no larger than ϵ. Since the only possible median bounds are bounds in $\mathcal{S}$, the median range will be a single alternative range in $\mathcal{S}$ after querying has progressed far enough.

The mechanism that selects the median most-preferred alternative from a finite set of alternatives is strategy-proof (dominant-strategy incentive compatible), and, as noted before, an elicitation algorithm that produces the same results as a dominant-strategy incentive compatible mechanism is ex-post incentive compatible. □

We emphasize that answering truthfully is an *exact* equilibrium, not an ϵ-equilibrium. The intuition behind the proof is the following. Effectively, if a partitioning query algorithm is used, then there is only a finite set of possible outcomes (alternative ranges). Due to SPPESNOR, each agent will have single-peaked preferences over these outcomes, and a query to an agent at a boundary point effectively asks the agent if her most-preferred outcome is to the left or right of this boundary point. When querying terminates, the final median bounding range is equal to the alternative range that is the median of the agents' most-preferred alternative ranges. Hence, a partitioning query algorithm under SPESSNOR preferences is effectively a simple elicitation algorithm for finding the median most-preferred alternative among a finite set of alternatives.

In the remainder of this section, we show how Theorem 5.1 can be applied to some basic elicitation algorithms. We first consider *half-half query algorithms*, which are algorithms that only use half-half queries (querying an agent i at $(\underline{x_i} + \overline{x_i})/2$), and that, once an agent's bounding range has become less than or equal to ϵ in size, never ask that agent another query (as is justified by Proposition 3.2).

Lemma 5.2 *If an agent initially has bounding range $B = [l, u]$, then the qth half-half query to the agent is at a point of the form $l + i(1/2)^q(u - l)$ (where i is an integer). Equivalently, after $q \geq 0$ half-half queries to the agent, each of the agent's bounds is at a point of the form $l + i(1/2)^q(u - l)$.*

The proof of this lemma is straightforward. Using this lemma, we can show that half-half query algorithms are partitioning query algorithms; hence, Theorem 5.1 applies.

Theorem 5.3 *If*

1. *all agents have SPPESNOR;*
2. *initially, every agent has the same bounding range $B = [l, u]$; and*
3. *agents are queried using a half-half query algorithm;*

then if we query until the median point is found to within ϵ, answering truthfully is an ex-post equilibrium.

Proof Let k be the smallest integer such that $(1/2)^k(u-l) \leq \epsilon$. Querying until the median point is found to within ϵ never requires more than k queries to an agent, because k half-half queries results in the agent having a bounding range of size $(1/2)^k(u-l) \leq \epsilon$, and if every agent has a range with size at most ϵ, then the median range has size at most ϵ, by Proposition 3.2.

Lemma 5.2 specifies that half-half query number q to some agent will be at a point of the form $l + j(1/2)^q(u-l)$, for some integer j. We can rewrite the form of the query points as

$$l + j(1/2)^q(u-l) = l + j(1/2)^{q-k}(1/2)^k(u-l)$$

Since l, u, and k are constant across all queries, and $j(1/2)^{q-k}$ is an integer (when $q \leq k$), the query points form a partition of the space of possible points. So, by Theorem 5.1, this query mechanism is ex-post incentive compatible. □

As it turns out, Lemma 5.2 can also help us show that median halving algorithms are, under certain circumstances, partitioning algorithms. Intuitively, each halving stage cuts the current median bounding range in half, and hence acts like a half-half query on the median bounding range; so, the points at which we query should have the same form as in half-half query algorithms. However, it can be the case that a halving stage cuts the current median bounding range by more than half, if initially some agents already have bounds inside the current median bounding range. If this is not the case, however, the median bounding range will be exactly halved:

Lemma 5.4 *Let the median bounding range be M. If no agents' bounds are within M except (possibly) at the midpoint of M, then a median halving algorithm will exactly halve M with each halving step, by moving one of the bounds of M to the midpoint of M, and no bounds will be inside the resulting new median bounding range.*

Proof Let h be the midpoint of M. If there are no agents' bounds inside M except (possibly) at h, a median bound cannot move to any point other than h. A median bound can only move to a point where an agent's bound lies, and the only such point inside M during a halving step is h, because the halving step asks queries only at h. □

Using this lemma, we can show that if all agents start with the same bounds, then median halving algorithms are partitioning algorithms, hence Theorem 5.1 applies.

Theorem 5.5 *If all agents have SPPESNOR, and, initially, every agent has the same bounding range $B = [l, u]$, then for any median halving algorithm, answering truthfully is an ex-post equilibrium.*

Proof Initially, since all agents have the same bounding range, no agents' bounds are within M. By Lemma 5.4, whenever queries cause a median bound to move, it moves to the current midpoint of the median bounding range. This is exactly the same as if the median bounding range were a single agent's bounding range, being repeatedly halved by half-half queries. So, Lemma 5.2 specifies that halving step q asks its queries at a point of the form $l+i(1/2)^q(u-l)$, for some integer i.

Since each halving step exactly halves the median bounding range, we require k halving steps to converge, where k is the smallest integer such that $(1/2)^k(u-l) \leq \epsilon$. Just as in the proof of Theorem 5.3, we can rewrite the form of the query points as

$$l + j(1/2)^{q-k}(1/2)^k(u-l)$$

When $q \leq k$, $j(1/2)^{q-k}$ is an integer, and the query points are of the form specified in Lemma 5.2. So, this query mechanism is ex-post incentive compatible. □

5.2 Median range elicitation incentives

We now return to the fully general setting where each agent i has a privately held most-preferred value range, $P_i = [l_i, u_i]$. We wish to elicit agents' most-preferred value ranges until we have found the median most-preferred range to within some tolerance ϵ. During elicitation we maintain bounds $\underline{l_i}, \overline{l_i}, \underline{u_i}, \overline{u_i}$ on each agent i's lower and upper bounds. We also maintain bounds $\underline{m_l}, \overline{m_l}, \underline{m_u}, \overline{m_u}$ on the median lower and upper bounds, and the final output of the elicitation mechanism consists of the final versions of these bounds, that is, the two ranges $[\underline{m_l}, \overline{m_l}]$ and $[\underline{m_u}, \overline{m_u}]$. We will elicit the median range to within ϵ, that is, $\overline{m_l} - \underline{m_l} \leq \epsilon$ and $\overline{m_u} - \underline{m_u} \leq \epsilon$.

The SPPESNOR definition no longer makes sense in this context, since the output is now given by two ranges rather than a single one. Informally, we generalize SPPESNOR to this setting by requiring that agents have SPPESNOR for the lower bound, and SPPESNOR for the upper bound (analogously to the definition of single-peaked preferences over ranges, which informally means that agents have single-peaked preferences over the lower bound and single-peaked preferences over the upper bound).

Definition 5.3 An agent i, whose most-preferred range is $P_i = [l_i, u_i]$, has *single-peaked preferences over pairs of equally-sized non-overlapping ranges (SPPESNOR2)* if, for any pairs of ranges $X = ([\underline{l}, \overline{l}], [\underline{u}, \overline{u}])$ and $X' = ([\underline{l'}, \overline{l'}], [\underline{u'}, \overline{u'}])$ for which SPPESNOR implies (given most-preferred point l_i) $[\underline{l}, \overline{l}] \succeq_i [\underline{l'}, \overline{l'}]$, and SPPESNOR implies (given most-preferred point u_i) $[\underline{u}, \overline{u}] \succeq_i [\underline{u'}, \overline{u'}]$, it is always the case that $X \succeq_i X'$.

In Sect. 4.2.2, we discussed how we can use a range bounding query to simulate a bound query on an agent's lower (or upper) bound, hence we can elicit lower and upper bounds separately. In that section, if a range bounding query that was intended to give information about the agent's lower bound also gave us some information about the upper bound, we happily made use of that information. However, to more easily examine incentives, we will now assume that when determining upper bounds, we do not make use of anything that we learned while querying for lower bounds. This leads to the following definition:

Definition 5.4 Let A be an elicitation algorithm that, in the setting where every agent i has a most-preferred value x_i, produces a range $[\underline{m}, \overline{m}]$ within which the median lies. The *two-stage version* of A operates in the setting where every agent i has a most-preferred range $P_i = [l_i, u_i]$, and first produces $[\underline{m_l}, \overline{m_l}]$ using A (by simulating queries on the agents' lower bounds using range bounding queries), and then produces $[\underline{m_u}, \overline{m_u}]$ using A (by simulating queries on the agents' upper bounds using range bounding queries, ignoring anything learned in the first stage).

Using this idea of a two-stage algorithm, we can take all the earlier algorithms that incentivize truthful answers under SPPESNOR (Theorems 5.1, 5.3, and 5.5), and turn them into algorithms that incentivize truthful answers under SPPESNOR2.

Theorem 5.6 *Suppose that, in the setting where every agent i has a most preferred value x_i, under algorithm A answering queries truthfully is an ex-post equilibrium if agents have SPPESNOR. Then, in the setting where every agent i has a most-preferred range $P_i = [l_i, u_i]$, under the two-stage version of A answering queries truthfully is an ex-post equilibrium if agents have SPPESNOR2.*

6 Features and partial instantiations of topics

So far, we have discussed eliciting one value range from each agent. One reason that allowing agents to submit ranges (rather than point values) can be useful is that a topic may not be completely specified. For example, consider the topic "Increase in temperature between 2009 and 2100." This topic is too imprecisely specified for even the ideal climate scientist to give a single number. Perhaps such a scientist could provide an exact estimate of the increase in temperature in Antarctica, given that a specific climate pact is adopted to reduce emissions of greenhouse gases; or of the increase on the North Pole, given that no climate pact is adopted at all. The region under consideration, as well as the change in human behavior, are examples of relevant *features* of the topic. If the scientist's projected increase is 2 degrees for the former scenario, and 6 degrees for the latter scenario, this still leaves her with a range of (at least) [2, 6] for the general (underspecified) topic. A solution may be to vote separately over each completely specified topic, but this is impractical: first, it is difficult to determine every feature of the topic that is relevant, and second, the number of completely specified topics is exponential in the number of features. Hence, agents must vote over underspecified topics; however, we may be interested in the effect of specifying at least some of the features.

Formally, we have a general *topic*, and a set F of *features* of the problem. Each $f \in F$ is defined by a set of values that the feature can take. For example, the feature region can take values {Antarctica, North Pole, Equator}. A *partial instantiation* π of a topic has zero or more features fixed at a certain value. For example, region = North Pole, climate pact = ? is a partial instantiation. A *complete instantiation* of a topic has all features fixed at values. Partial instantiation π is *more specific* than π' if π has all the features that are fixed in π' fixed to the same value (and *strictly more specific* if π has at least one additional feature fixed). Each agent i has a value range $P_{i,\pi}$ for each partial instantiation π (so for a given partial instantiation, we can use the median rule, as before). We assume that if π is more specific than π', then $P_{i,\pi} \subseteq P_{i,\pi'}$. We also assume that if $P_{i,\pi} = [l_i(\pi), u_i(\pi)]$, then there exist complete instantiations π', π'' with $l_i(\pi) = l_i(\pi')$ and $u_i(\pi) = u_i(\pi'')$ (we will refer to this assumption as the *spanning assumption*). We allow nondegenerate ranges even for complete instantiations.

The set of all partial instantiations is generally too unwieldy. It is more natural to arrange some of the partial instantiations of a topic into a *tree*. Every node v of the tree corresponds to a partial instantiation of the topic, π_v. The root of the tree has no features instantiated. A node's children always correspond to strictly more specific partial instantiations; moreover, all of a node's children have exactly one more feature instantiated, this feature is the same for all children, and every value for this feature corresponds to some child. The tree's leaves may or may not be complete instantiations; if they are, we say that the tree is *complete*. Figure 8 gives an example.

If we want to elicit an agent's ranges for the partial instantiations in the tree, one natural way of doing so is to first query the agent for her range at the root node, then for the ranges at its children, *etc.* (We note that our use of the word "query" here is different from the use in Sects. 4 and 5: here, a query immediately produces the entire range. If desired, such a query can be implemented by asking subqueries in the manner of Sects. 4 and 5.) One advantage of this is that any range for a node must contain the range of each of its descendants. Figure 8 illustrates this.

6.1 Finding good trees

For any topic with more than one feature, we can construct several different trees, each containing a different subset of the topic's partial instantiations. For example, Fig. 8 shows one

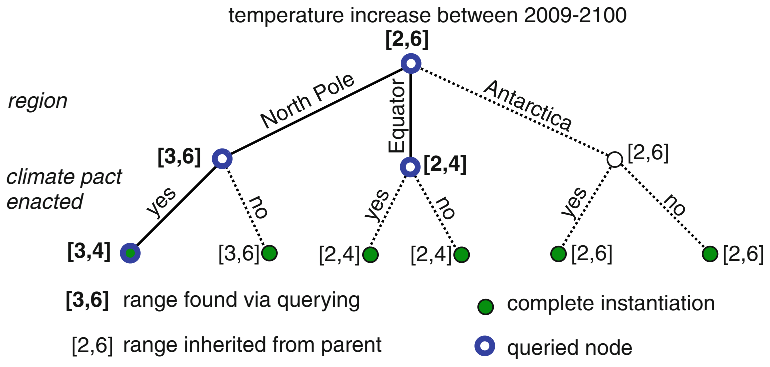

Fig. 8 An example tree with ranges for the partial instantiations

elicitation tree for the climate change topic, with the feature region being queried before the feature climate pact enacted, but one could easily imagine a different tree that queried these features in reverse order.

We would like the features that are most relevant to the agent's range to be close to the root of the tree, for the following reason. As the remaining unspecified features become less relevant, the ranges that the agent reports will shrink, perhaps to a point. Thus, if the relevant features are close to the root, then at relatively shallow nodes the reported ranges will have magnitude at most (say) ϵ; the nodes deeper in the tree can then inherit these ranges and be reasonably well approximated. As we will show, we can then use this to approximate the range for *any* partial instantiation (even ones that are not in the tree).

A *frontier node* is a node that has been queried, and that is either a leaf or the closest queried ancestor to some unqueried node. (For example, in Fig. 8, the root is a frontier node because it has a child on the right that has not been queried.)

Lemma 6.1 *If for some tree, every leaf is the descendant of a frontier node (this is always true if querying started at the root), and the magnitude of the range at each frontier node is less than ϵ, then for every complete instantiation π, we can infer a range X_π with magnitude less than ϵ within which the true range P_π must lie.*

Proof We can always extend the tree to a complete tree. Then, by the definition of a frontier node, each complete instantiation can inherit the range from some frontier node, which has magnitude less than ϵ. □

We can use these bounding ranges X_π for the complete instantiations to approximate the range for *any* partial instantiation.

Lemma 6.2 *If every complete instantiation π has a bounding range X_π of magnitude less than ϵ, then for any partial instantiation π' (not necessarily in the tree), we can infer a range $X_{\pi'}$ within which the true range $P_{\pi'}$ must lie, and whose magnitude is at most 2ϵ larger than that of $P_{\pi'}$.*

Proof Let $C(\pi')$ be the set of complete instantiations that are more specific than π'. Then, let $X_{\pi'}$ be the smallest range that contains $\bigcup_{\pi \in C(\pi')} X_\pi$. This range has the desired properties, for the following reasons. For the *true* ranges P_π, it must be the case that the smallest range that contains $\bigcup_{\pi \in C(\pi')} P_\pi$ is equal to $P_{\pi'}$, by the spanning assumption. For each $\pi \in C(\pi')$, we have $P_\pi \subseteq X_\pi$, and the magnitude of X_π exceeds that of P_π by at most ϵ. Therefore, the smallest range that contains $\bigcup_{\pi \in C(\pi')} X_\pi$ must contain the smallest range that contains $\bigcup_{\pi \in C(\pi')} P_\pi$, and its magnitude can be at most 2ϵ larger (since we can have error of up to ϵ on each side). □

Theorem 6.3 *If for some tree, the magnitude of the range at each frontier node is less than ϵ, then for any partial instantiation π' (not necessarily in the tree), we can infer a range $X_{\pi'}$ within which the true range $P_{\pi'}$ must lie, and whose magnitude is at most 2ϵ larger than that of $P_{\pi'}$.*

Proof The theorem follows immediately from Lemmas 6.1 and 6.2. □

From this, it follows that we can use different trees for different agents, because, given that the frontier nodes' ranges become sufficiently small, we can closely approximate each agent's range for *any* partial instantiation. Let us now study how to find a good tree for a single agent.

Rather than first designing a tree for an agent, and then eliciting the agent's ranges, it can be helpful to *dynamically* construct the tree during the elicitation process. For example, we can, for each feature separately, instantiate only that feature and query the agent about the resulting partial instantiations. This way we can decide which feature is most effective in shrinking the agent's ranges, place it at the top of the tree, and continue. This greedy algorithm is similar to choosing the feature that maximizes *information gain* in decision trees. In general, this will not produce the optimal tree: for example, two features may be very helpful when considered together, but useless individually, in which case the greedy algorithm will not use them. Nevertheless, one would expect the greedy algorithm to do well in practice.

We now present the algorithm formally. In the algorithm, v is the current node of the tree (which initially is the root). π is the current partial instantiation (which initially has nothing instantiated). F' is the set of remaining features (initialized to F). $\iota(\pi, f \leftarrow w)$ is the partial instantiation obtained by starting with π and then instantiating feature f to value w. W_f is the set of values that feature f can take. The algorithm proceeds as follows:

1. For every feature $f \in F'$, let $S_f = \{\iota(\pi, f \leftarrow w) : w \in W_f\}$ (the set of partial instantiations that result from instantiating f).
2. For every $f \in F'$, for every $\pi' \in S_f$, query the agent to obtain her preferred range $P_{\pi'}$.
3. For every $f \in F'$, compute a heuristic $h(\{P_{\pi'} : \pi' \in S_f\})$ indicating how effective f is in reducing the ranges. For example, one might use $h(\{P_{\pi'} : \pi' \in S_f\}) = \frac{1}{|S_f|} \sum_{\pi' \in S_f} |P_{\pi'}|$, the average range size after branching on the feature. Let f^* be a feature that minimizes the heuristic.
4. Branch on f^*. For every value $w \in W_{f^*}$, create a child node $v_{f^* \leftarrow w}$, and recurse with $v_{new} = v_{f^* \leftarrow w}$, $\pi_{new} = \iota(\pi, f^* \leftarrow w)$, and $F'_{new} = F' - \{f^*\}$.

Even this greedy algorithm typically asks a large number of queries. The number of queries can be further reduced by sorting all features by h at the root, and then simply branching in this order (at depth k, branch on the $(k+1)$th-best feature). We have implemented this version, together with a visual interface, as a Web applet, which is available at http://www.cs.duke.edu/~jfarfel/ranges/. In the future, we hope to use this tool on some real-world problems—for example, letting climatologists use it to determine reasonable ranges of temperature increase. Would climatologists using these tools arrive at generally accepted ranges for temperature increases, or completely different ones?

6.2 Finding good median trees

In general, as output, we would like to produce a tree with the median ranges. One way of doing this is to elicit each agent's preferences separately, leading to potentially different trees for the agents; however, because such a tree also allows us to find an approximation of the agent's range for any partial instantiation not in her tree, we can still use this information to construct a single tree of median ranges.

Alternatively, we can use a modified version of the single-agent tree elicitation algorithm to find a good tree of median ranges, as follows. Where the single-agent algorithm queries for the agent's range at a node, we instead query for the median range at the node, by asking all agents for their ranges at this node. (Alternatively, we can use, say, a median halving algorithm to obtain the (approximate) median range at the node.) At each step, the heuristic h computes how effective feature f is in reducing the median ranges, and we branch on the feature that minimizes h.

There are a few issues with this approach. First, it requires that all agents are accessible for querying throughout the construction of the tree of median ranges, since every time we query for the (approximate) median range, we may have to query any given agent (in the worst case). In contrast, if we use the single-agent algorithm, we can use an online approach where, once an agent arrives, we elicit its complete preferences, after which the agent is free to leave. Second, if we approximate the median range, there may be ramifications to how we do this. For example, in order to get an approximation at some step in the process, we could use a median range halving algorithm, which outputs two ranges, $M_l = [\underline{l_m}, \overline{l_m}]$ and $M_u = [\underline{u_m}, \overline{u_m}]$, bounding the median lower and upper bounds to within ϵ. There are several feasible approximations of the median range that we can derive from this output—for instance, we could use the largest feasible approximating range, $[\underline{l_m}, \overline{u_m}]$, or the smallest one, $[\overline{l_m}, \underline{u_m}]$. The nature of the approximation we choose may affect how "good" the tree of medians turns out to be.

7 Conclusions

We studied settings in which multiple agents each have a most-preferred range of values, and an aggregate range must be chosen. For example, the agents may be climatologists trying to determine a range for the increase in temperature between 2009 and 2100; legislators determining a range for the prison term for a robbery; reviewers determining a range for the quality of a paper; etc. In such settings, an agent may wish to misreport her most-preferred range so that the aggregate range better matches her own. To prevent this, the rule for selecting the aggregate range must be *strategy-proof*, that is, no agent should have an incentive to misreport. When agents submit points rather than ranges, a well-known rule is to select the *median* submitted point. This rule is strategy-proof if preferences are *single-peaked*. We generalized the definition of single-peaked preferences to ranges, and proved that a generalized median rule—choose the median of the submitted lower (upper) bounds as the aggregate lower (upper) bound—is strategy-proof for such preferences.

We also studied how to elicit median points and ranges approximately, using bound queries and a generalization called range bounding queries. We introduced a class of elicitation algorithms called median halving algorithms, and analyzed the number of queries these algorithms require. Eliciting a median point or range only approximately can affect agents' incentives to answer queries truthfully, but we gave sufficient conditions for when an elicitation algorithm is such that answering queries truthfully is an (exact) ex-post equilibrium (and median halving algorithms satisfy these conditions). Finally, we considered the possibility that specifying additional features of the setting (such as the region in which the temperature increase is measured, or whether the robbery was armed) can allow agents to refine their ranges. We developed a framework and algorithms for selectively choosing which features to specify.

Acknowledgments We thank the reviewers for helpful comments. This work is supported by NSF Grant IIS-0812113 and an Alfred P. Sloan Fellowship.

References

1. Barberà, S., Massó, J., & Serizawa, S. (1998). Strategy-proof voting on compact ranges. *Games and Economic Behavior, 25*(2), 272–291.
2. Black, D. (1948). On the rationale of group decision-making. *Journal of Political Economy, 56*(1), 23–34.
3. Border, K. C., & Jordan, J. S. (1983). Straightforward elections, unanimity and phantom voters. *The Review of Economic Studies, 50*(1), 153–170.
4. Boutilier, C. (2002). A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th national conference on artificial intelligence (AAAI)* (pp. 239–246). Edmonton, Canada.
5. Braziunas, D., & Boutilier, C. (2005). Local utility elicitation in GAI models. In *Proceedings of the 21st annual conference on uncertainty in artificial intelligence (UAI)* (pp. 42–49). Edinburgh, UK.
6. Chajewska, U., Getoor, L., Norman, J., & Shahar, Y. (1998). Utility elicitation as a classification problem. In *Proceedings of the 14th conference on uncertainty in artificial intelligence (UAI)* (pp. 79–88). Madison, WI, USA.
7. Chajewska, U., Koller, D., & Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *Proceedings of the 17th national conference on artificial intelligence (AAAI)* (pp. 363–369). Austin, TX.
8. Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research, 35*, 161–191.
9. Conitzer, V., & Sandholm, T. (2002). Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the 18th national conference on artificial intelligence (AAAI)* (pp. 392–397). Edmonton, Canada.
10. Conitzer, V., & Sandholm, T. (2005). Communication complexity of common voting rules. In *Proceedings of the sixth ACM conference on electronic commerce (EC)* (pp. 78–87). Vancouver, Canada.
11. Gibbard, A. (1973). Manipulation of voting schemes: A general result. *Econometrica, 41*, 587–602.
12. Pini, M. S., Rossi, F., Venable, K. B., & Walsh, T. (2005). Aggregating partially ordered preferences: Possibility and impossibility results. In *Proceedings of the tenth conference on theoretical aspects of rationality and knowledge (TARK)*, Singapore.
13. Pini, M. S., Rossi, F., Venable, K. B., & Walsh T. (2007). Incompleteness and incomparability in preference aggregation. In *Proceedings of the 20th international joint conference on artificial intelligence (IJCAI)*, Hyderabad, India.
14. Rossi, F., Pini, M. S., Venable, K. B., & Walsh, T. (2006). Strategic voting when aggregating partially ordered preferences. In *Proceedings of the fifth international conference on autonomous agents and multi-agent systems (AAMAS)* (pp. 685–687). Hakodate, Japan.
15. Sandholm, T., & Boutilier, C. (2006). Preference elicitation in combinatorial auctions. In P. Cramton, Y. Shoham, & R. Steinberg (Eds.), *Combinatorial auctions.* (Chap. 10) (pp. 233–263). Cambridge: MIT Press.
16. Satterthwaite, M. (1975). Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory, 10*, 187–217.
17. Vu, H., & Haddawy, P. (1997). Problem-focused incremental elicitation of multi-attribute utility models. In *Proceedings of the 13th conference on uncertainty in artificial intelligence (UAI)* (pp. 215–222). San Francisco, CA, USA.
18. Vu, H., & Haddawy, P. (1998) Towards case-based preference elicitation: Similarity measures on preference structures. In *Proceedings of the 14th conference on uncertainty in artificial intelligence (UAI)* (pp. 193–201).
19. Xia, L., & Conitzer, V. (2008). Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the 23rd national conference on artificial intelligence (AAAI)* (pp. 196–201). Chicago, IL, USA.