Chapter 1
The Complexity of Manipulative Actions in Single-Peaked Societies

Edith Hemaspaandra · Lane A. Hemaspaandra · Jörg Rothe✉

Anna and Belle, who now have become so aware of their role as guides in this book that they can even refer to the book's content, meet on what will be a very exciting day for them. Let's listen in on their conversation.

"I have two pieces of wonderful news," says Belle.
"Tell me, tell me," says Anna.
"First, as shown in the previous chapter, many types of manipulative attacks on elections are computationally intractable. Like *wow*!"
"I'm glad that you're excited by that, but it isn't doing much for me. What is the second piece of good news?"
"Today is the annual charity Pumpkin Pie Taste-Off! You remember it well, I'm sure. Tables and tables of pumpkin pies are set out, and are compared based on their taste, competing for the coveted honor of being chosen as a best-tasting pumpkin pie."
"Like ***wow***! That is my favorite event of the entire fall season. I love pumpkin pie, at least when it tastes just right. But it is hard to get it just right. As everyone knows, the key is getting the sweetness level to be exactly right."
"Absolutely. Everyone I've ever met agrees that the way to judge pumpkin pies is by their level of sweetness. We seem to be in perfect agreement, as we usually are."

Anna and Belle rush down to the Pumpkin Pie Taste-Off, a yearly charity event of their town (see Figure 1.1). They find that there are 26 amateur and professional bakers competing in this year's contest. Each baker has brought a large, fresh-baked pumpkin pie, so that people can taste the 26 pies and then cast their votes—by a strict, linear ordering of the 26, of course! By a remarkable coincidence, it turns out that this year the 26 bakers have the last names Adams, Brown, Chavez, Dylan, . . ., Young, Zimmerman. By an even more amazing coincidence, it turns out that Adams baked the least sweet pie,

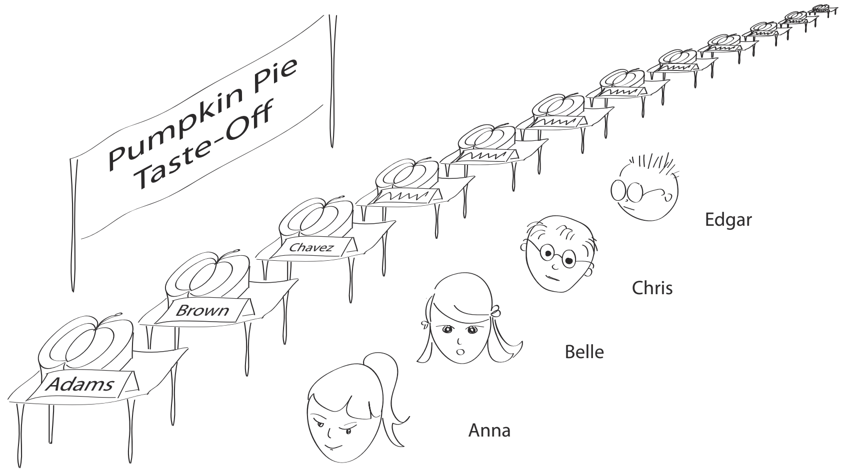


Fig. 1.1 Anna, Belle, Chris, and Edgar at the annual charity Pumpkin Pie Taste-Off

Brown baked the second least sweet pie, and so on through the alphabet up to Zimmerman, who baked the sweetest of all the pies.

It sounds as if this contest might be a run-away for Zimmerman, but let us listen in some more. By now, Anna, Belle, all their friends and family, and many others have tasted all the pies and voted.

Anna, Belle, Chris, and Edgar all say, simultaneously, “Well, that was an easy decision. My vote was based on the most important thing about pumpkin pies, their level of sweetness.”

“That is great,” says Anna. “Clearly we all gave our top spot for Zimmerman, whose pies are the sweetest; yummy! So my vote was Zimmerman $>$ Young $> \cdots >$ Dylan $>$ Chavez $>$ Brown $>$ Adams, of course—the only reasonable vote.”

“Now hold on a minute,” says Belle. “When I said that what matters about pumpkin pies is their level of sweetness, I obviously meant that the sweetness level had to be not too sweet and not too tart. Among these 26 entrants, King’s pie is the one that best matches, to my taste, that point of perfect balance. And to my taste, if one has to miss that point of perfect balance, it is better to miss on the side of being overly sweet, although not by too much. That is why, after tasting all the pies, my own vote put King first, Larsen second, Martinez third, Norton fourth, and Juarez fifth.”

“Taste must run in families,” says Edgar, Belle’s sister. “Like Belle, King’s pie to me is the ideal one. But big sis is wrong about which ones come right after that. To my taste, if one has to miss the perfect balance point, it is better to miss on the side of being a bit less sweet, although not by too much. That is why the top spots on my ballot went to, in this order, King, Juarez, Iverson, Heck, Larsen, and Gilchrist."

"You've all got it wrong," says Edgar's friend Chris. "It is Thibodeaux's pie that gets the sweetness just right. I gave it the top spot in my vote."

"Well, this is a fine mess," comments Belle. "Even though we all are rating pie based on their sweetness, we each have differing views on what the ideal level of sweetness is. And we also have different views as to how our liking for pies drops off as they diverge from our ideal sweetness point in one direction or the other, although for each of us, clearly between two pies that are sweeter than our ideal sweetness point, we'll prefer the one that is the less sweet of the two, and similarly, between two pies that are less sweet than our ideal sweetness point, we'll prefer the one that is the sweeter of the two."

Everyone answers, "Certainly, there is no doubt about that."

After short pause, in which they all think about this, Belle, who is very smart, exclaims, "Oh no! My first piece of wonderful news was that many types of manipulative actions on elections are computationally intractable. But our discussion of pies now has me very worried as to whether that is truly so. My worry is this: The intractability proofs were based on constructions that allowed arbitrary collections of voters. But we've just seen here that some electorates may cast only certain patterns of votes. For example, does anyone here think that someone might cast a vote that put Zimmerman first and Adams second?"

Everyone replies in unison, "Impossible! Unthinkable! No one with the ability to taste food could possibly cast such a bizarre vote."

"So," continues Belle, "perhaps the complexity of manipulative actions on elections that have restrictions on what votes can be cast—or on what collections of votes can be cast—might be far lower than the complexity is regarding the case where there are no restrictions on what votes can be cast? Perhaps those intractability results that so raised my spirits may turn to dust in this case, leaving the elections open to perfect polynomial-time attack algorithms? These delicious pieces of pie are costing me quite a bit of my peace of mind!"

Belle and her friends (whose preferences are graphically displayed as Figure 1.2) have touched upon a tremendously important point. As Belle realized, it is possible that if the collections of votes that can be cast are restricted, the complexity of manipulative attack problems may change. In this chapter, we will explore this for the case of the most famous and important restriction on electorate behavior. This restriction is known as *single-peaked* electorates, and it essentially is just the type of situation that Belle and her friends have innocently stumbled upon as they discussed the nature of their preferences regarding pumpkin pies.

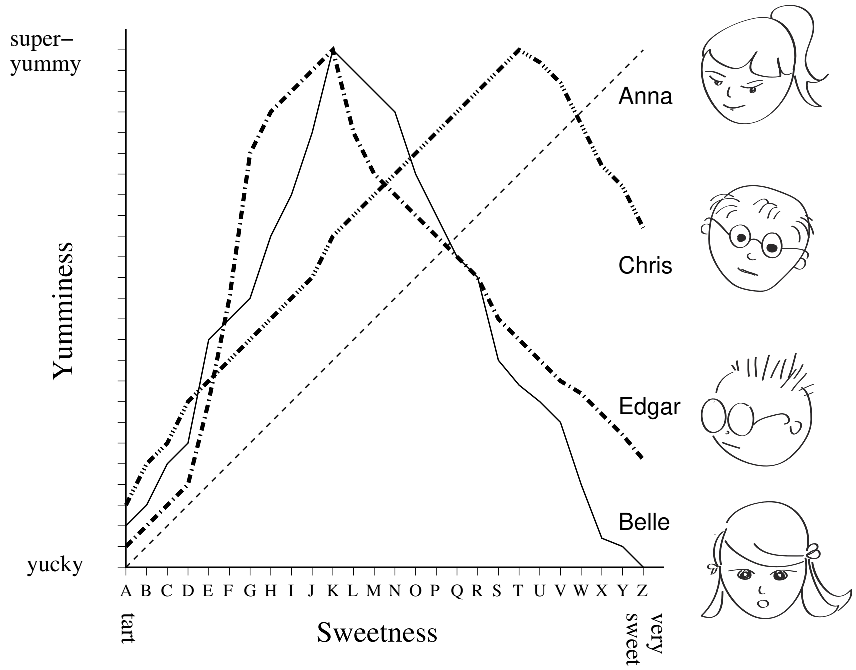


Fig. 1.2 Preferences regarding sweetness of pumpkin pie

We will see that single-peakedness often does lower the complexity of attacks on elections, just as Belle feared it would. However, we'll see that sometimes single-peakedness does not change the complexity of attacks on elections. And we will even see that, although this might seem so obviously "impossible" that Belle above did not even imagine that it could happen, there are cases where looking at the special case of single-peakedness increases the complexity of attacks on elections.

Our study of single-peaked elections will be structured as follows. Section 1.1 will more formally define single-peaked electorates, will discuss and further motivate them, and will mention how the study of single-peakedness is integrated into the key manipulative-action problems that were introduced in the previous chapter. Sections 1.2, 1.3, and 1.4 will cover some examples of control, manipulation, and bribery in the context of single-peaked electorates. Finally, Section 1.5 will hear from Helena, who has very surprising preferences regarding pumpkin pie. This will lead us to more generally consider what happens in electorates that are *nearly single-peaked*. That is, they may contain a few "maverick" voters who vote in ways potentially having nothing to do with the single-peakedness of the setting, e.g., voters who judge pumpkin pies based on the crust or the color of the filling. We will see that in some cases, the presence of even one such maverick can make the complexity of manipulative action problems jump back up to intractability.

1.1 Single-Peaked Electorates

We will now more formally define single-peaked electorates (for both voting by preferences and voting by approval vectors—recall these notions from Section **??**), will discuss single-peakedness and further motivate it, and will explain how single-peakedness can be integrated into the key manipulative-action problems that were presented in Section **??** to model control, manipulation, and bribery scenarios.

Black [3, 4] introduced the notion of single-peaked preferences in order to model societies that are heavily focused on a single issue, such as the level of sweetness in the Pumpkin Pie Taste-Off described above. Clearly, in the political world there often is a dominant issue on which the electorate is heavily focused and on which voter preferences are naturally single-peaked, be it level of taxation, breadth of the social welfare network, or degree of participation in an overseas military action. Even when there is no one salient issue, political parties as well as politicians themselves can often be linearly ordered according to their position on a left-right spectrum, where left-wing (right-wing) parties/politicians take a more liberal (conservative) position. Thus it is not at all surprising that single-peakedness is one of the key concepts of political science, and is central in the study of elections. Gailmard, Patty, and Penn, who studied Arrow's impossibility theorem (see Theorem **??** on page **??**) in the context of single-peaked electorates, described single-peakedness as "*the* canonical setting for models of political institutions" [21].

We now formally define this notion both for electorates whose preferences are linear rankings and for electorates using approval vectors.

Definition 1.1 (single-peaked preferences). Let C be a set of candidates.

1. A list V of votes over C, each vote in V being a linear order $>_i$, is said to be *single-peaked* if there exists a linear order L over C (which we will refer to as the *societal axis*) such that for each triple of candidates, a, b, and c, if $a\,L\,b\,L\,c$ or $c\,L\,b\,L\,a$, then for each i it holds that $a >_i b$ implies $b >_i c$.
2. A list V of approval vectors over C is said to be *single-peaked* if there exists a linear order L over C such that for each triple of candidates, a, b, and c, if $a\,L\,b\,L\,c$ then whenever a vote in V approves of a and c, it must also approve of b.

Anna wants to know, “What does this mean, actually?”

“It means,” explains Edgar, “that whenever you take any three candidates who are ordered consistently with the societal axis (like, for example, Adams, King, Larsen or Larsen, King, Adams in Figure 1.2), then in each individual vote, whenever the middle candidate of the three is ranked below one candidate, it must be ranked above the other candidate.”

“I still don’t get it!”

“Rule of thumb: *‘Never rank the middle candidate last!’* For example, among Adams, King, and Larsen in their societal order of Figure 1.2, if one of us were to put Larsen first, Adams second, and King last, then we wouldn’t be single-peaked with respect to this societal axis. This is because if you prefer, say, Larsen to King, just as you and Chris do in Figure 1.2, Definition 1.1 *requires* you to prefer King to Adams. On the other hand, it is absolutely fine to prefer King to *both* Adams and Larsen, as Belle and I do; that doesn’t contradict Definition 1.1. And remember, that applies to *all* triples of candidates, not just to Adams, King, and Larsen, and it also applies to *each* of us voters.”

“Another way to put it is,” Belle adds, “that for each of us, with respect to the societal axis, our preference-based utilities rise to a peak and then fall, or just rise, or just fall. That is why it is called *single*-peaked. For example, Anna, your preferences in Figure 1.2 ‘just rise.’ ”

“If we aren’t single-peaked with respect to some given societal axis (like the alphabetical order of Figure 1.2), does this mean we cannot be single-peaked at all with our preferences?”

“No,” Belle replies, “there might be another societal axis for which our preferences indeed are single-peaked. All that matters is that there *exists* at least one such axis. Actually, I wonder how difficult it is to find out whether a given list of votes, as linear rankings, in fact are single-peaked. After all, there are $m!$ ways to order m candidates on a societal axis, and that is a huge number of possible axes to check!”

“That’s easy!” Edgar claims. “Give me your list of votes and I’ll tell you whether they are single-peaked in no time at all.”

Edgar is right that this is an easy problem in the sense that it can be solved efficiently (though not “in no time at all,” as he claims, but rather in *polynomial time*—recall the foundations of complexity theory outlined in Section **??**). Indeed, Bartholdi and Trick [1] show that, given a list of linear rankings over the candidates, it can be decided in polynomial time whether they are single-peaked, and that when they are, one can also find one societal axis—in fact, even (in implicit form) *all* societal axes—witnessing the single-peakedness. They show this by transforming this problem in polynomial time into the problem of determining whether a matrix has the so-called “consecutive ones property.” The result then follows from the work from Fulkerson and

Gross [20, Sections 5 and 6] and Booth and Lueker [6, Theorem 6]. Doignon and Falmagne [9] (see also the work of Escoffier, Lang, and Öztürk [13], Fitzsimmons and Lackner [19], and Elkind, Lackner, and Peters [11]) give a direct (and faster) polynomial-time algorithm for this problem.

"But what about a given list of approval vectors?" Anna then asks. "What does single-peakedness mean in that case?"

"As a rule of thumb, this simply means: *'Never leave a gap in your approvals!'* Of course, this again refers to a societal axis that works for the complete list of approval vectors. When a voter goes along the societal axis, say from left to right, and approves of a first candidate, the voter may then keep approving of further candidates, but as soon as the voter next disapproves of anyone, the voter can't go back to approving. That is, there is just a single peak consisting of a contiguous (possibly empty) interval of approved candidates. Pretty simple!"

"One could also say," Belle adds, "that with respect to the societal axis, we each rise to a peak where we approve and then fall back to disapproval, or we always approve, or never approve. But all this talk makes me wonder how difficult it is to find out whether a given list of approval vectors in fact is single-peaked."

"That's easy, too!" Edgar exclaims. "Give me your list of approval vectors, and I'll tell you whether they are single-peaked in no time at all."

Again, Edgar is right. As pointed out by Faliszewski et al. [17, Section 2], the work of Fulkerson and Gross [20, Sections 5 and 6] and Booth and Lueker [6, Theorem 6] shows that, given a list of approval vectors, it can be decided in polynomial time (in fact, in a certain natural sense even in linear time) whether they are single-peaked, and if so, one can also find one societal axis—in fact, even (in implicit form) *all* societal axes—witnessing the single-peakedness. In effect, testing whether a given list of approval vectors is single-peaked is the same as testing whether a matrix whose columns are those approval vectors has the consecutive ones property.

In the following sections we will study problems modeling control, manipulation, and bribery scenarios—recall these notions from the previous chapter, in particular from Sections **??**, **??**, and **??**—when restricted to single-peaked electorates. In each of these restricted problem variants, it is important to note that a societal axis L witnessing the single-peakedness of the given electorate is part of the problem instance. (So inputs that don't contain a valid such axis are not "Yes" instances of the given problem.) Also, it is important to note that the electorates both *before* and *after* the manipulative action must be single-peaked with respect to that same—i.e., the given—societal axis L. For example, in the single-peaked restrictions of control problems such as constructive control by adding voters (CCAV, as defined on page **??**

in Section **??**), we require that the entire list of votes, including even votes of any unregistered voters, be single-peaked with respect to L. That strong requirement itself immediately ensures single-peakedness both before and after this control action. We will denote the single-peaked restriction of CCAV by SP-CCAV, etc. For manipulation problems (where votes are being specified) and bribery problems (where votes are being outright changed), we similarly require that both the initial vote set and the final vote set be single-peaked with respect to the axis L that was provided as part of the problem's input.

1.2 Control of Single-Peaked Electorates

Let us recall Belle's insightful comment.

"Perhaps the complexity of manipulative actions on elections that have restrictions on what votes can be cast—or on what collections of votes can be cast—might be far lower than the complexity is regarding the case where there are no restrictions on what votes can be cast? Perhaps those intractability results that so raised my spirits may turn to dust in this case, leaving the elections open to perfect polynomial-time attack algorithms? These delicious pieces of pie are costing me quite a bit of my peace of mind!"—Belle

Let us start right in, with a theorem showing precisely this, for an important voting system, and what is probably the most important type of control. To make clear why the following theorem really is showing a case where single-peakedness reduces complexity (unless $\mathrm{P} = \mathrm{NP}$), it is important to keep in mind that constructive control by adding voters for approval voting is NP-complete (see Table **??** on page **??** in Section **??**).

Theorem 1.1 (Faliszewski et al. [17]). *For the single-peaked case, approval voting is vulnerable to constructive control by adding voters.*

The above result holds in both the nonunique-winner model and the unique-winner model, and holds both in the standard model of input, in which each vote appears on a distinct ballot, and in the so-called "succinct" input model, in which the input is a list of the distinct preference orders cast by at least one voter and each such preference order is paired with a nonnegative integer coded in binary that indicates how many voters cast that preference as their vote.

We won't include a full, formal proof of this theorem. Rather, using an extended example we'll convey the idea of the proof. In particular, in the example we'll show how the polynomial-time algorithm for this problem works. In our example, we'll use the nonunique-winner model and the succinct input model (although for better readability, in Figures 1.3 through 1.9 we'll write numbers in base 10 rather than in binary).

While Belle relaxes to restore her peace of mind, our guides in the following extended example will be our emerging experts on single-peakedness, Anna and Edgar.

"I think I do understand Theorem 1.1," says Anna, "even though I'm not sure *why* it actually holds. Do you have any idea, Edgar? Would you please clarify this all for me?"

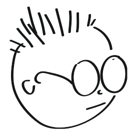
"Sure," Edgar replies, "it's easy. We can establish the theorem by giving a polynomial-time algorithm solving the constructive-control-by-adding-voters problem for approval voting in the single-peaked case. Now, what's the input to our algorithm?"

"I know, I know!" Anna exclaims. "According to the problem definition of CCAV on page **??** in Section **??**, we are given some votes (which are approval vectors) over the candidates, including our preferred candidate p. And we have some additional votes (again approval vectors). We know that all votes are single-peaked with respect to the given societal axis—also part of the input. And our goal will be to check whether we can make p a winner by adding less than or equal to a certain number of the additional votes; that number, the so-called addition limit, itself is also given as part of the problem."

Edgar says approvingly, "Exactly! We want to solve the problem SP-CCAV for approval voting. Let's assume that we are in the succinct input model, so the input doesn't explicitly list all the votes but only has, written down in binary, how often each vote occurs that has been cast at least once. This compact representation can only make it harder for our algorithm to run in time polynomial in the input size, so if our algorithm runs in polynomial time in the succinct case, it surely also does so in the case of the standard input model."

"But how does the algorithm work?"

Edgar replies, "I'll give you an example. Let us look at Figure 1.3!"



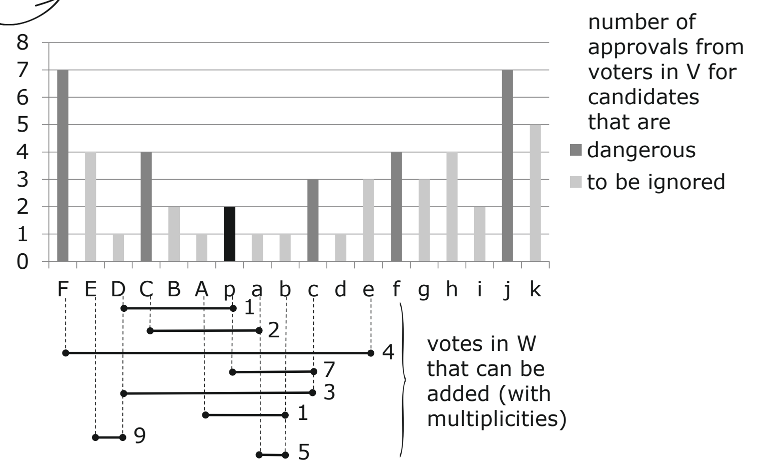


Fig. 1.3 Proving Theorem 1.1 by example: input to SP-CCAV

"What you see here," Edgar explains, "are 18 candidates ordered from left to right along our societal axis, F, E, D, C, B, and A to the left of p, then p, and then $a, b, \ldots, k$ to the right of p. The diagram shows the number of approvals each candidate has from the already registered voters (those in V); for example, j has seven approvals but p has only two. And as mentioned above, keep in mind that the input must also give us the limit on how many votes we may add."

"Why are some of the candidates called *'dangerous,'* while others are *'to be ignored'*?" interrupts Anna.

"Wait a minute, and I'll answer that later. First, do you know why the votes from W—those that may be added (and each coming with a number saying how often it occurs)—are all intervals? For example, note the two votes approving of C, B, A, p, and a?"

"Of course, I know that!" says Anna, brimming with indignation. "It is because they are single-peaked approval vectors. So they cannot have gaps between their approvals!" She pauses to ponder for a second. "But, which types of vote should we add to the election, especially if two votes are *incomparable*? For example, if you look at the two votes approving of C through a versus the seven votes approving of p through c in Figure 1.4: Both vote types approve of p, and that is great. But they also approve of different other candidates. The former but not the latter helps p relative to b and c; the latter but not the former helps p relative to C, B, and A. If we have lots of such choices to make, I foresee a combinatorially explosive number of collections of votes that we need to consider as possible choices for the set of votes we should add, and so I would not be surprised if this whole thing ended up being NP-complete. Or do you have some clever way of avoiding that combinatorial explosion, by wisely deciding which should be added?"

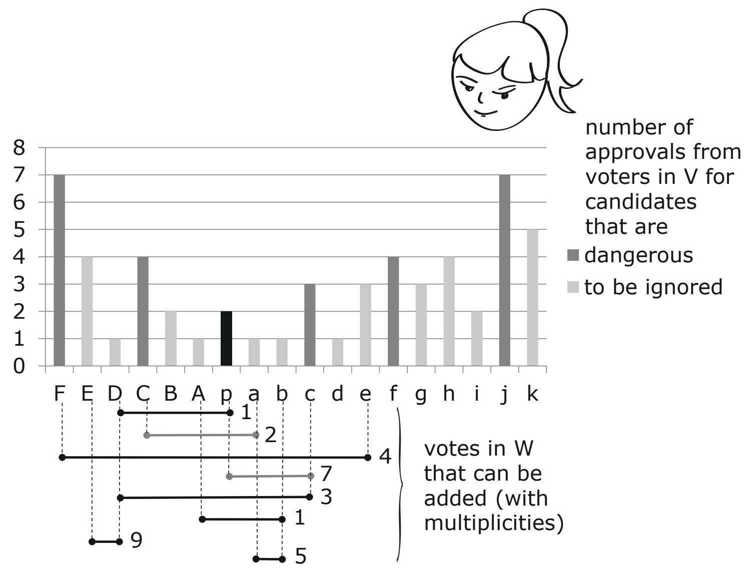


Fig. 1.4 Proving Theorem 1.1 by example: two incomparable votes

"You have identified the heart of the matter! But have no fear. We'll handle this—and avoid any combinatorial explosion—by a 'smart greedy' algorithm letting us make such choices in a decisive way that assures us that if either of the choices can lead to success, then the choice we make will lead to success," says Edgar. "I'll explain that algorithm later."

"You always postpone answering my questions!" Anna is not amused. "A minute ago you similarly avoided explaining to me why some of the candidates are called 'dangerous,' namely F, C, c, f, and j, while all others are 'to be ignored.' So please tell me now why you have labeled them in these ways!"

"OK. First, each added vote of course will be an interval including p; it would be insane to add votes whose interval does not include p. So we drop all other votes. Figure 1.5 shows the result of doing so in our example: The nine votes approving of only E and D and the five votes approving of only a and b have been dropped. All remaining votes include p.

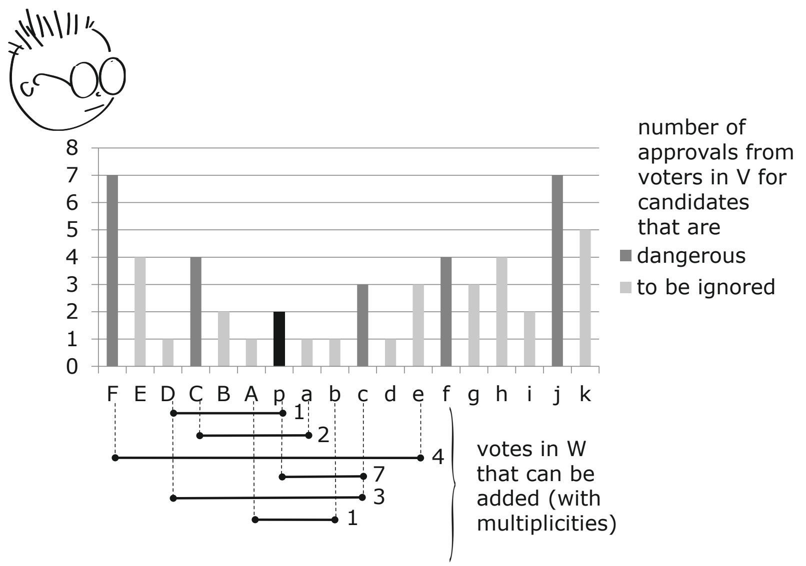


Fig. 1.5 Proving Theorem 1.1 by example: dropping all votes not approving of p

"That doesn't tell me why, for instance, c is a 'dangerous' candidate," says Anna.

"Well, if adding votes from (what remains of) W causes p to draw level with c in terms of approvals, then—since all remaining votes include p—p must at least draw level with a and b. That is since, due to our interval property, every vote that approves of p and c also approves of a and b, yet p as you can see starts this part of our algorithm with at least as many approvals as a and b."

"Agreed."

"Thus c is a dangerous rival for p, and a and b can safely be ignored. Likewise, f is dangerous for p but d and e can safely be ignored. And similarly, j is dangerous for p but g, h, and i can safely be ignored."

"Hey, why do you do that step by step? Just say j is dangerous for p, and ignore a, b, c, d, e, f, g, h, and i! Figure 1.6 shows what I mean, both for the candidates to the left and to the right of p. And while I'm complaining, let me mention that although a and b started with fewer approvals than p, e for example starts with more, and so the reasoning you applied above as to why a and b can be ignored seems to me to not apply to e."

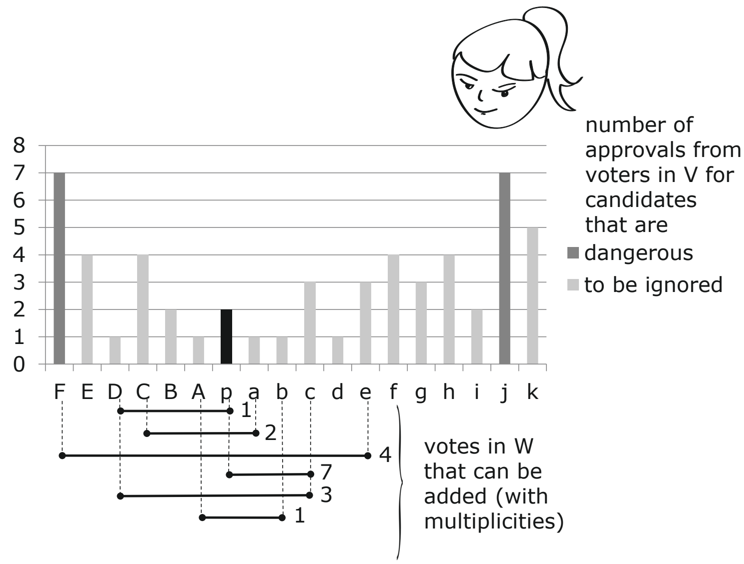
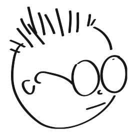


Fig. 1.6 Proving Theorem 1.1 by example: Anna suggests to not go step by step

"The two points you just made are deeply intertwined, as both are connected to the importance of this algorithm going step by step," Edgar says as, startled, he jumps over to Figure 1.7. "Let us consider your suggestion that we say that j is the only dangerous candidate to the right of p. Look what happens if we add, say, five of the seven votes approving of p through c. Then the number of these votes in W is reduced to two, and each of p, a, b, and c get five more approvals. No doubt, p has now drawn level with j, both having seven approvals now, but c was riding the wave and was boosted to even eight approvals! So if we don't go step by step, p might well draw level with j but still is not a winner."



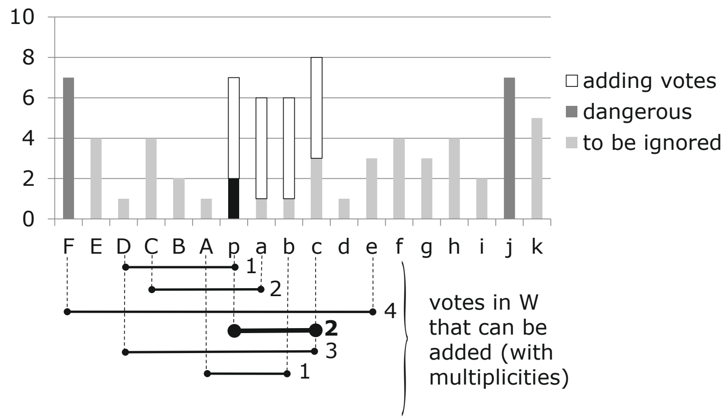


Fig. 1.7 Proving Theorem 1.1 by example: What happens if we don't go step by step?

"I see," Anna concedes. "That means the first dangerous candidate to the right of p is the leftmost candidate to the right of p that is approved by more voters from V than p, namely c in the figure, and the second dangerous candidate to the right of p is the leftmost candidate to the right of c that is approved by more voters from V than c, namely f, and so on. And we can define this analogously for the candidates to the left of p. So, as indicated in Figure 1.8, we indeed get the dangerous candidates C and F to the left and c, f, and j to the right of p, and we ignore all other candidates. And knowing this, I understand the right answer to my second point above—the one where I pointed out that e for example starts with more approvals than p. I now see that that is true but, crucially, *after* we have made p tie or beat c in approvals—by adding only votes that approve of p but don't approve of c (and thus also don't approve of e!)—at *that* point p will surely tie or beat all candidates to c's right that started with no more approvals than c. So f indeed is the next dangerous candidate; e is not a worry at all!"

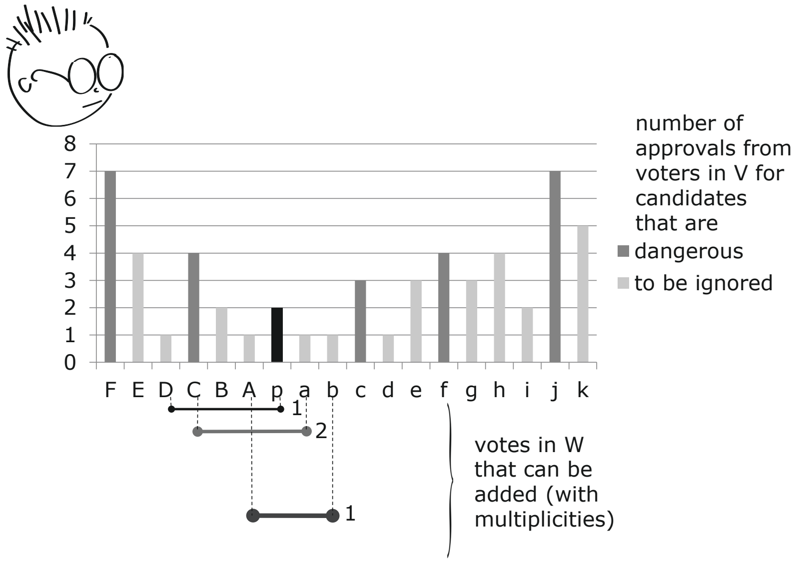


Fig. 1.8 Proving Theorem 1.1 by example: Which votes can help in "smart greedy"?

"Exactly! Of course, what we have just been discussing is the notion of *dangerous candidate for p* in the nonunique-winner model. But what change do you think we'd want to make if we are in the unique-winner model?"

"Then I'd say, when looking for the first dangerous candidate on for example the right side of p, we take whichever candidate is the first to have *at least as many approvals* as p, all else being the same."

"That's right," Edgar agrees. "If we want to make p a *unique* winner, even having the same number of approvals as p already makes a candidate a dangerous rival for p; we must ensure that p strictly beats this candidate. That is, B too would be dangerous for p in the example of our figure.

Anna nods her agreement, and then suggests, "However, let's stay in the nonunique-winner model, which seems to be more natural. Now tell me, how does your 'smart greedy' algorithm work? How does it find the right votes from W to add?

"In the 'smart greedy' algorithm, we need to eat through all dangerous rivals to the right of p, starting with the leftmost, c. To become a winner, p in particular must draw level with c. However, only votes (i.e., intervals) in W whose right endpoints fall into $[p, c)$ can help."

"I see. These are exactly the votes from W that still are shown in Figure 1.8."

"Let X be the set of these votes," Edgar continues. "Now, the key insight of the algorithm is that we will be choosing votes from X

starting with those having the rightmost left endpoint. In our example of Figure 1.9, we start by adding the voter—shown by a fat line—approving of A through b. As shown in Figure 1.9, this is already enough for p to draw level with c, so p's first dangerous rival has been taken care of. And the key point is that this has been achieved by an easy (in the sense of polynomial time) yet perfectly safe strategy!"

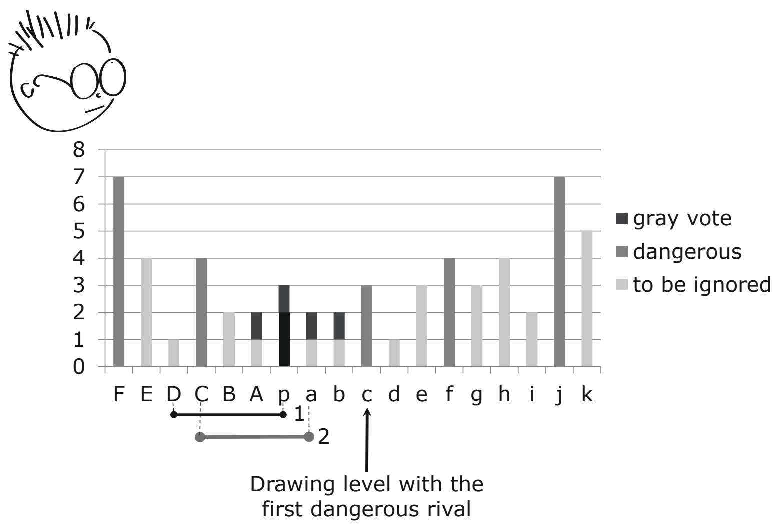


Fig. 1.9 Proving Theorem 1.1 by example: drawing level with the first dangerous rival

"Why?" asks Anna.

"Because if there is *any* way at all to choose votes to add such that p draws level with his first dangerous rival (and eventually can become a winner), our strategy of starting with those votes from X that have the rightmost left endpoint will succeed, too. (And if there is no such way, then of course this strategy cannot succeed either.) And among all such choices, note that crucially our choice is at least as good regarding how it leaves us relative to other dangerous candidates—most crucially those to the left of p—since it is approving of as few of them as possible. Briefly put, the extra candidates on the right that the 'fat' vote helped do us no harm, and the extra candidates on the left that the nonfat votes would have helped might do us harm. (By the way, before moving on, I should mention that if we had two or more copies of the fat vote, and p needed at that point two or more approvals relative to c, then we would in a single step have added as many copies of the fat vote as needed, or if that is more than the number of copies that we had available, then would have added all its copies—unless either of those took us beyond our addition limit, in which case we'd have to admit defeat. The reason I mention this is to make clear that we really are handling the succinct case—and so we can't add votes one at a time, but rather add them drawing on the right 'multiplicity' of the given vote to support our progress.)"

"Now that p's first dangerous rival has been taken care of," says Anna, "what do we do next?"

"We iterate. That is, updating the approvals of all candidates and the votes that may be added as in Figure 1.9, we apply the same procedure to handle the next dangerous candidate, here f, as long as our allowed number of votes to be added hasn't been used up. If we run out of dangerous candidates on the right-hand side of p, we reverse the societal order, and we finish off the remaining dangerous candidates (which have been mirrored from the left to the right of p to make the same procedure applicable to them) in exactly the same way until we either succeed in making p a winner and so can output 'yes,' or reach the addition limit without having achieved our goal. In the latter case, we know for sure that no strategy whatsoever could possibly make p win by adding the allowed number of votes from W, so we can safely output 'no,' i.e., that success is not possible."

That concludes our extended example sketching the polynomial-time algorithm for constructive control by adding voters under approval voting when one is dealing with a single-peaked electorate.

But Anna can be a tough person to convince of anything.

Anna: Thank you for that example. I do believe your polynomial-time claim for constructive control by adding voters in the single-peaked case. But I worry: Maybe that is the only control type where single-peakedness helps, and maybe approval is the only voting system showing this behavior.

Edgar: Have no fear, they are not alone!

The following result gives some examples of what Edgar is referring to—other control cases that are NP-complete in the general case but have polynomial-time algorithms for the case of single-peaked electorates. Recall the definitions of these control problems from Section **??** starting on page **??**.

Theorem 1.2 (Faliszewski et al. [17]).

1. *For the single-peaked case, approval voting is vulnerable to constructive control by deleting voters.*
2. *For the single-peaked case, plurality voting is vulnerable to constructive and destructive control by adding candidates, by adding an unlimited number of candidates, and by deleting candidates.*

Having seen this theorem, which we mention holds in both the nonunique-winner model and the unique-winner model, Anna has been convinced—but perhaps a bit *too* convinced, as the following shows.

Anna: Wow. Those additional cases make it clear to me that restricting our focus to single-peaked electorates lowers the complexity. I'll bet that this approach will undercut all existing NP-hardness result regarding all election problems.

Edgar: Not so fast! In fact, for the devilishly complex system STV (which is defined on page **??** in Section **??**), Walsh [25] has noted that even when restricted to single-peaked electorates, the possible winner problem remains NP-complete and the necessary winner problem remains coNP-complete (see Section **??** for the problem definitions); sometimes, hard things stay hard even under single-peaked preferences.

1.3 Manipulation of Single-Peaked Electorates

Our guides Anna and Belle (the latter of whom has through resting recovered her peace of mind) are chatting again, and the chat takes a shocking turn.

Anna: In our discussion of control, we saw that restricting our focus to single-peaked electorates sometimes lowers the complexity. And Edgar mentioned to me an example, regarding possible and necessary winners, where restricting our focus to single-peaked electorates fails to lower the complexity. Clearly, that covers all the possibilities, since restricting ourselves to single-peaked electorates obviously cannot ever raise the complexity.

Belle: I disagree. I claim that restricting ourselves to single-peaked electorates *can* raise the complexity!

Anna: That's clearly not possible. Anyone who knows the basics of complexity knows that if a problem is easy, any easily identified restricted case of it is also easy. In this case, since every single-peaked electorate is an electorate, it follows that if the problem has a polynomial-time algorithm for all electorates, then it has a polynomial-time algorithm for single-peaked electorates.

Belle: No, my dear friend. I understand what you're thinking, and your error is a quite subtle one. The error is hidden in your words "the problem" above. You are assuming that "the problem" is the same in both cases. If that were true, your claim about subcases inheriting polynomial-time algorithms would be fine. But the problems in question do *not* differ merely on whether the electorate must be single-peaked.

Anna: I don't see any other way in which they differ.

Belle: That is where the "subtle" comes in. Recall that in defining our problems in the single-peaked context, we required that the electorates be single-peaked (with respect to the given societal axis) even *after* the manipulation.

Anna: Yes, that is natural, but what does it have to do with some difference in the problems.

Belle: The difference is that for the single-peaked case of manipulation, we are asking whether (the input is single-peaked with respect to the societal axis L and) there is some set of votes by the manipulators *under which the election is still single-peaked with respect to the axis* L and p is a winner of the election. In contrast, the general-case is merely asking whether there is some set of votes by the manipulators such that p is a winner of the election.

Anna: Then the single-peaked case gives fewer options to the manipulators as to what votes they can cast, and so the problem is a subcase, and so as I said before it can only be simpler.

Belle: No. It is a different problem. The "subcases only reduce complexity" argument line only refers to restrictions of the problem domain. If the actions inside the problem can differ, even if they are more restrictive, that is a whole different issue. It is possible that for the less restrictive set of actions a manipulation problem is computationally easy, even though it would be computationally hard for a more restrictive set of actions, such as being limited to manipulations that leave the electorate single-peaked.

Anna: Huh?

Belle: Let me try to give you a bit of intuition as to how this might happen. Let us consider constructive size-3-coalitional unweighted manipulation, and our model will be that votes are approval vectors. However, our voting system won't be approval voting. In fact, suppose our voting system has the property that when the electorate isn't single-peaked, then it is easy to manipulate successfully. As an extreme example, consider a voting system that when the electorate isn't single-peaked makes all candidates be winners, and thus makes the preferred candidate p be a winner; and if the electorate is single-peaked this system chooses some winner in some different and very complex-to-manipulate way. So a coalition of three or more manipulators can achieve success—even if we for a moment jump out of the model where the societal axis is fixed and given—simply by having each of the three ballots (which among themselves form a "Condorcet-cycle"-like pattern) $a > b > c$, $b > c > a$, and $c > a > b$ (where a, b, and c represent the names of the three lexicographically smallest candidates) be cast by at least one manipulator. And if the number of candidates is less than three, we in our election system just have everyone always win. Note that there is no axis that makes any vote collection with the just mentioned three votes be single-peaked. Clearly, manipulation for this problem is in polynomial time for the general case. But the single-peaked case can't use this approach, since it isn't allowed to manipulate votes in such a way as to violate single-peakedness; and in fact, the single-peaked case has no such easy, obvious path to successful manipulation. Indeed, one can specify a voting system of this sort in such a way that the manipulation problem for the single-peaked case is NP-complete.

Anna: I certainly don't see all the details, since you didn't specify them, but I do see the general flavor. Your example counterintuitively makes single-peakedness's more limited set of legal electorates a complexity-increasing disadvantage, rather than a complexity-lowering advantage. And I see that that isn't paradoxical, because the single-peaked case not only limits the set of legal inputs, but also limits the set of legal manipulative actions the coalition can take. So we aren't merely a special case of a problem; we're a slightly different problem, since the single-peakedness in some sense penetrates the problem to its very core.

Belle: Well put; I see a real future for you as a complexity theorist when we grow up.

Anna: Heaven forfend! Anyway, I've always thought that your little brother Edgar was the most likely of any of us to live *that* life.

Belle: Heaven forfend! He's already insufferable enough as it is, and I hear that complexity theorists are beyond insufferable.

Anna: I've heard that too.

The theorem that Belle was outlining is the following result. Its detailed proof, which we won't give here, is a bit twisty, especially regarding achieving the NP-completeness part. But Belle's description of the proof strategy is in fact spot-on.

Theorem 1.3 (Faliszewski et al. [17]). *There exists a voting system $\mathcal{E}$, whose votes are approval vectors, for which constructive size-*3*-coalition unweighted manipulation is in polynomial time for the general case but is* NP-*complete in the single-peaked case.*

Anna: But now that I'm at least dabbling at thinking at things complexity-theoretically, I'm wondering whether that strange complexity-raising behavior can ever happen for systems that I'm familiar with. In particular, if you can show me any scoring protocol under which we get this complexity-raising behavior, I'll give you my next ten slices of pumpkin pie.

Belle: I love pie but, alas and alack, I cannot show you an example. But neither can anyone else, since no such example can exist!

Let us see what Belle—who despite her protestations seems well on her way to becoming a complexity theorist—is referring to. Recall from Chapter **??** that for scoring protocols (defined in Section **??**) there is a dichotomy theorem (stated in Section **??**) for the constructive coalitional weighted manipulation problem. In particular, we mentioned there what in effect is the following result.

Theorem 1.4 (Hemaspaandra and Hemaspaandra [22]). *For each m and each scoring protocol $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$, the constructive coalitional weighted manipulation problem is* NP*-complete if $\alpha_2 > \alpha_m$, and is in* P *otherwise.*

For scoring protocols, there also is a dichotomy theorem for the constructive coalitional weighted manipulation problem in the single-peaked case.

Theorem 1.5 (Brandt et al. [7]). *Consider an m-candidate scoring protocol $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$.*

1. *If $m \geq 2$ and $\alpha_2 > \alpha_{\lfloor (m-1)/2 \rfloor + 2}$ and there exist integers $i > 1$ and $j > 1$ such that $i + j \leq m + 1$ and $(\alpha_1 - \alpha_i)(\alpha_1 - \alpha_j) > (\alpha_i - \alpha_{i+1})(\alpha_j - \alpha_{j+1})$, then the constructive coalitional weighted manipulation problem for the single-peaked case is* NP*-complete.*
2. *If $m \geq 2$ and $\alpha_2 = \alpha_{\lfloor (m-1)/2 \rfloor + 2}$ and $\alpha_1 > \alpha_2 > \alpha_m$ and ($\alpha_2 > \alpha_{m-1}$ or $\alpha_1 - \alpha_m > 2(\alpha_2 - \alpha_m)$), then the constructive coalitional weighted manipulation problem for the single-peaked case is* NP*-complete.*
3. *In all other cases, the constructive coalitional weighted manipulation problem for the single-peaked case is in* P*.*

The above result, Theorem 1.5, truly is a dichotomy theorem; it proves that every scoring protocol is either NP-complete or in P. In fact, this dichotomy theorem even has a very easy-to-check condition that tells for a given scoring protocol which case holds for it, thus making this theorem very easy to apply to actual, natural protocols. For example, from the theorem we can immediately see that for each m it holds that for the single-peaked case of the constructive coalitional weighted manipulation problem, m-candidate plurality and m-candidate veto are in P; and so is m-candidate Borda for $m < 4$. The theorem also makes clear that m-candidate Borda is NP-complete for each $m \geq 4$. Given this theorem's ease of application, one probably cannot fairly complain about how very much more involved the theorem statement is than the analogous and also easy-to-apply theorem for the general case, Theorem 1.4. If anything, one should—while thanking the universe for the fact that the characterization is easy to apply—blame the universe for making the single-peaked case have such a complex-looking characterization. What isn't complex to observe is what Belle was commenting about: Every P case from the general-case dichotomy of Theorem 1.4 clearly remains a P case in the single-peaked dichotomy of Theorem 1.5. So, just as Belle claimed, in the setting of scoring protocols and constructive coalitional weighted manipulation, restricting attention to the single-peaked case never raises complexity. As to the NP-complete cases from the general-case dichotomy of Theorem 1.4, in the single-peaked dichotomy of Theorem 1.5 some of those cases remain NP-complete and some fall to P.

Belle: I've now shown you why no example of the sort you requested can exist.

Anna: True, but since you didn't provide the requested example, I'm still keeping those ten future slices of yummy pie.

Belle: Grrrrr!

Theorem 1.5 hides, underneath its complexity, a quite broad range of cases. And some of those at first are surprising. For example, the behavior that people typically expect for voting systems, viewed at each fixed number of candidates, is that as one increases the number of candidates, the problem either stays the same in complexity or increases in complexity. But let us apply Theorem 1.5 to the case of 3-veto. Recall that the scoring vector for 3-veto is given by $(1,\ldots,1,0,0,0)$; for example, for $m=5$ candidates this is $(1,1,0,0,0)$ and for $m=6$ candidates it is $(1,1,1,0,0,0)$. We get the following strange behavior, which was first noticed and proven by Faliszewski et al. [17].

Theorem 1.6 (Faliszewski et al. [17]). *For the single-peaked case, the constructive coalitional weighted manipulation problem for m-candidate 3-veto is in* P *for $m \in \{3,4,6,7,8,\ldots\}$, and is* NP*-complete for $m=5$.*

So this is a case where, between $m=5$ and $m=6$, the complexity *drops*. Let us get a sense of how this kind of unexpected behavior can arise. 3-veto for $m=3$ is in effect triviality, and so everyone always wins. 3-veto for $m=4$ is just plurality (1-approval), and the single-peakedness is irrelevant since every 1-approval vote is trivially single-peaked. So these two cases are certainly in P.

The $m=5$ case is shown NP-hard by a standard type of reduction, which we will now describe, again in the nonunique-winner model. (Membership of the problem in NP is obvious, so we in fact have NP-completeness once NP-hardness is shown.) To prove NP-hardness, we will give a reduction from the NP-complete problem Partition, which has been defined on page **??** in

Section **??** as the set of all nonempty sequences $(k_1,\ldots,k_n)$ of positive integers summing up to an even number, $2K=\sum_{i=1}^{n} k_i$, that can be partitioned into two subsequences, each summing up to the same value K. Suppose, we are given an input $(k_1,\ldots,k_n)$ of PARTITION, and for concreteness let us say we have $n=3$ and we will consider two particular cases for illustration:

1. $(k_1,k_2,k_3)=(1,2,3)$, so $K=3$. Since $1+2=3$, this is a yes-instance of PARTITION.
2. $(k_1,k_2,k_3)=(1,2,5)$, so $K=4$. Note that this is a no-instance of PARTITION.

(Of course, just handling how to reduce from these very special instances of PARTITION does not establish NP-hardness, since these two cases are quite trivial. However, although we will use these cases as illustrations, we in fact will be quietly giving the general case of this reduction.)

We construct an instance of the constructive coalitional weighted manipulation problem from this PARTITION instance as follows. Our candidates are Adams, Brown, Chavez, Dylan, and the preferred candidate Pearl that the manipulators wish to make a winner. We also fix the following societal axis L:

$$\text{Chavez } L \text{ Adams } L \text{ Pearl } L \text{ Brown } L \text{ Dylan.}$$

There are two nonmanipulators with weight K each, Anna and Belle, and Anna votes

$$\text{Chavez} >_{\text{Anna}} \text{Adams} >_{\text{Anna}} \text{Pearl} >_{\text{Anna}} \text{Brown} >_{\text{Anna}} \text{Dylan},$$

while Belle votes

$$\text{Dylan} >_{\text{Belle}} \text{Brown} >_{\text{Belle}} \text{Pearl} >_{\text{Belle}} \text{Adams} >_{\text{Belle}} \text{Chavez}.$$

As is sometimes the case for best friends, Anna and Belle have completely opposite preferences.[1] Obviously, both votes are single-peaked with respect to L, with Anna's preference-based utility "just falling" and Belle's preference-based utility "just rising." In addition, there are n manipulators where the ith manipulator has weight k_i; in our example with $n=3$ manipulators, Chris has weight k_1, David weight k_2, and Edgar weight k_3. The manipulative boys want to see Pearl win.[2] We will now show that they can reach their goal by suitably setting their preferences if and only if $(k_1,\ldots,k_n)$ is a yes-instance of PARTITION. In particular, they can ensure Pearl's victory in the first case $((k_1,k_2,k_3)=(1,2,3))$, but not in the second $((k_1,k_2,k_3)=(1,2,5))$.

---

[1] They are not voting here on who bakes the best pumpkin pie, but rather, let us say, on which of the bakers is the most beautiful person—again, tastes differ.

[2] Because Pearl promised them pumpkin pie for life if Pearl wins. But that bribe notwithstanding, the computational problem here still is about manipulation; bribery will be handled in the next section.

Suppose there is a successful partition of $(k_1, \ldots, k_n)$, i.e., suppose we have a set $A \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in A} k_i = \sum_{i \in \{1,\ldots,n\} \setminus A} k_i = K$ (e.g., $1+2=3$ as in our first case). Then all manipulators whose weight is in the set $\{k_i \mid i \in A\}$ (so Chris and David in our first case) can set their preferences to be

$$\text{Pearl} >_i \text{Adams} >_i \text{Brown} >_i \text{Chavez} >_i \text{Dylan},$$

while all other manipulators (namely, Edgar in our first case) can choose the preference order

$$\text{Pearl} >_i \text{Brown} >_i \text{Adams} >_i \text{Chavez} >_i \text{Dylan}.$$

Note that these manipulative votes are single-peaked with respect to L. Recall that for $m = 5$ candidates our scoring vector is $(1,1,0,0,0)$. Thus, in the election with both the manipulative and the nonmanipulative votes, Adams, Brown, and Pearl each score $2K = 6$ points, while Chavez and Dylan get only $K = 3$ points each, so Pearl is a winner.

For the converse, suppose now that there is no partition of $(k_1, \ldots, k_n)$ (such as in our second case). Can the manipulators, who are obliged to cast single-peaked votes with respect to L, still make Pearl a winner? Seeking a contradiction, let us assume that the answer is yes. Note that in each such single-peaked vote, whenever Pearl scores a point, Adams or Brown does so also. Thus, among the manipulative votes, Pearl's score is bounded above by the sum of the scores of Adams and Brown. Note that Pearl doesn't get any points from the nonmanipulators, but Adams and Brown receive K points each from them (in our example, Adams gets K points from Anna and Brown gets K points from Belle). Since we assumed that Pearl wins the election whose voter set includes both the manipulators and the nonmanipulators, it follows that among the manipulative votes alone, Pearl must score at least K points more than Adams does and at least K points more than Brown does. It follows that from the manipulative votes Pearl's score is $2K$, and Adams and Brown score K points each. However, this implies that the weights of the manipulators ranking Adams in their top two positions sum to K (and the same applies to the manipulators ranking Brown in their top two positions), that is, there is a partition of $(k_1, \ldots, k_n)$, a contradiction. This concludes our informal proof that our reduction for the $m = 5$ case is correct. Thus the constructive coalitional weighted manipulation problem is NP-hard.

Anna: Wait a minute! What if we are in the unique-winner model? Does this reduction apply to that case, too?

Belle: Almost. Just change the weights of the two nonmanipulators from K to $K - 1$.

Turning now to a discussion of the $m > 5$ cases in Theorem 1.6, we get the at-first-surprising drop in complexity. However, we claim that that drop has a quite clear source. Let's consider the $m = 7$ case, to see how it can possibly be simpler than the $m = 5$ case. Note that for $m = 7$ and 3-veto voting, each vote cast is a 4-approval vote. So whichever candidate is the middle (i.e., fourth) one among the 7 candidates, along the societal axis L, certainly must be approved of by every voter, since the votes are single-peaked, and thus each vote's approved-of candidates must be contiguous within L. So that middle candidate is always a winner. And each other candidate, a, is a winner exactly if every voter approves of a. Since we can certainly make all the manipulators approve of a, the only issue we need to look at to efficiently decide whether a can be made a winner is whether all nonmanipulative voters approve of a. We have just given a polynomial-time algorithm for the constructive coalitional weighted manipulation problem for 7-candidate 3-veto, in the single-peaked case. And the algorithm makes clear the type of effect at issue here—an effect that clearly will hold for all $m > 7$ also. Namely, we have that one or more candidates are forced to be approved of by every voter, and so the only real issue in this problem is whether a given candidate is approved of by all nonmanipulative voters. One can also argue that the $m = 6$ case is in P, though one has to be a bit more careful.

1.4 Bribery of Single-Peaked Electorates

In our study of control, we saw that some problems that are NP-complete in the general case fall all the way down to polynomial time for the single-peaked case. Where this behavior can be found depends on the exact setting: what the manipulative action is and what the voting system is.

For bribery, we have similarly mixed behavior. Under the single-peaked restriction, some NP-complete bribery cases fall from being NP-complete to being in P, and some do not. Whether such a complexity drop occurs is sensitive to both the type of bribery and the voting system.

Let us give some examples of these behaviors. For approval voting, the following theorem gives three cases where types of bribery that are known (some by Faliszewski, Hemaspaandra, and Hemaspaandra [14] and some by Brandt et al. [7]) to be NP-complete in the general case fall into polynomial time for the single-peaked case. Before looking at the theorem, recall the various notions of bribery defined in Section **??** starting on page **??**; in particular, the basic problem variant Bribery, which can be weighted (indicated by ⚖ in the problem name) and/or priced (indicated by \$), and the notions of negative and strongnegative bribery.

Theorem 1.7 (Brandt et al. [7]).

1. *For the single-peaked case, approval-Bribery is in* P*.*
2. *For the single-peaked case, approval-Negative-Bribery is in* P*.*
3. *For the single-peaked case, approval-Strongnegative-Bribery is in* P*.*

Of course, that is just one voting system. Can we cast a wider net as to understanding when bribery problems fall in polynomial time for the single-peaked case? The answer is a "yes," although as we'll soon see, it is a somewhat qualified "yes."

Recall the notion of weak Condorcet winner from page **??** in Section **??**. In particular, recall that a voting system is said to be *weakCondorcet-consistent* if whenever there are candidates that tie-or-beat all other candidates in head-on-head pairwise contests, the winner set is exactly the set of candidates having that property. It turns out that for the single-peaked case we can in one fell swoop capture the bribery complexity, under all eight standard types of bribery, of all weakCondorcet-consistent voting systems. Five of the bribery types are simple and three are complex.

Theorem 1.8 (Brandt et al. [7]). *Let $\mathcal{E}$ be any voting system that is weakCondorcet-consistent, or even that merely is always weakCondorcet-consistent on single-peaked electorates.*

1. *For the single-peaked case, $\mathcal{E}$-⚖-\$Bribery, $\mathcal{E}$-Negative-⚖-Bribery, and $\mathcal{E}$-Negative-⚖-\$Bribery are each* NP*-complete.*
2. *For the single-peaked case, $\mathcal{E}$-Bribery, $\mathcal{E}$-\$Bribery, $\mathcal{E}$-⚖-Bribery, $\mathcal{E}$-Negative-Bribery, and $\mathcal{E}$-Negative-\$Bribery are each in polynomial time.*

Many important systems are weakCondorcet-consistent. For example, the voting systems—each defined in Section **??**—Llull, maximin, Young, weak-Dodgson (by weak we mean Dodgson altered so that the goal is to by adjacent-exchanges make a candidate tie-or-beat each other candidate in head-on-head contests), and weakBlack (with weak analogously interpreted) are weakCondorcet-consistent [18, 7]. And as noted by Brandt et al. [7], the voting systems of Kemeny, Schwartz, Llull, and two variants of Nanson are weakCondorcet-consistent when restricted to single-peaked electorates. So Theorem 1.8 applies to all the just-mentioned systems, and classifies all its types of bribery.

Are any of the P results obtained in that way examples of complexity being reduced due to single-peakedness? Absolutely. For example, for Llull elections, bribery, \$Bribery, ⚖-Bribery, and ⚖-\$Bribery, are each NP-complete [16], but by Theorem 1.8 each of these cases is in P for the single-peaked setting. For Kemeny the drop is even more dramatic. Each of the eight standard types of bribery is $\mathrm{P}^{\mathrm{NP}}_{||}$-hard for Kemeny elections [7]. Yet by Theorem 1.8, in the single-peaked case three of those $\mathrm{P}^{\mathrm{NP}}_{||}$-hardness bounds change to NP-completeness results and five change to P results.

On the other hand, let us come back to our earlier comment about the "yes" regarding the wider net being a qualified "yes." What we meant by that is that part of what is underpinning Theorem 1.8 is the fact that in single-peaked electorates (with the voters voting by linear orders), there always is a weak Condorcet winner, namely, a candidate who ties-or-beats each other candidate.

And that means that all the voting systems we are discussing here (and more generally, all weakCondorcet-consistent voting systems) become the same as each other for the case of single-peaked electorates, namely, the winner set in that case for each is exactly the collection of all weak Condorcet winners. On one hand, that might be viewed as disappointing, since the systems are all becoming the same system, for the case that Theorem 1.8 is speaking of. On the other hand, the more interesting and important points to focus on are how very varied those systems are in the general case and how hard their bribery problems often are in the general case, and yet despite that how single-peakedness removes so many of those hardness results for those systems.

1.5 Do Nearly Single-Peaked Electorates Restore Intractability?

Let us return to the Pumpkin Pie Taste-Off. Although everyone seemed to be in agreement that level of sweetness was what anyone who had the ability to taste food uses to judge pumpkin pies, a surprising twist is about to occur.

"Did I just hear you say that no one with the ability to taste food could possibly put Zimmerman first and Adams second, given that Zimmerman makes the sweetest pumpkin pie, Adams makes the least sweet, and the twenty-four other contestants' pies fall between them in sweetness," says a quiet voice from off to the side, belonging to Helena. "That in fact is exactly what I would have cast as my vote had I been here in time to vote."

"Impossible! Unthinkable! Have you no sense of taste?"

"I do," replies Helena, "but I have celiac disease, and so energetically avoid eating the protein known as gluten. And only the pies of Zimmerman and Adams are gluten-free, thanks to the ingredients in their crusts being made respective of rice flour and amaranth flour. When I judge a pie, the crust's ingredient set is the issue that I use."

Quick-witted Belle, who is still aware that she is helping guide us through this book, immediately says, "You may just have made my day, you wonderful maverick! As I mentioned earlier, I was overjoyed that complexity might provide a shield against attacks on elections. Then when pumpkin pie led us to discuss single-peaked elections, I became worried that in that natural setting the protections might evaporate. The past few sections of this book in large part showed my fears to be well-grounded. We've seen that single-peakedness often does sidestep existing complexity-theoretic protections against attacks on elections. But you have opened my eyes to a new hope. Those sidestepping results were based on the assumption that the electorates in question are single-peaked. However, it seems to me that pretty much no electorate will be perfectly single-peaked. There will always be at least a few mavericks. At this Taste-Off, although we all thought it obvious that every person judges pumpkin pie based on the sweetness, we found that you judge based on the crust. And in large political elections where almost everyone is voting based on the candidates' positions on some important spectrum—perhaps liberal versus conservative—there surely will be a few people who see things differently. Perhaps some people are libertarians and so care about an aspect not even captured by that axis, and perhaps others are influenced by issues such as a candidate's religion or a candidate's charisma. It seems to me at least possible that for cases where there are some mavericks amidst a largely single-peaked society, some of the complexity-theoretic protections against manipulative attacks may still hold. At the very least this is worth looking into ... although only after I reward myself with another slice of King's pumpkin pie!"

Belle makes an excellent point, and in this section we'll see examples of the type of behavior that she is imagining. Indeed, we'll even see that in some cases, the presence of *a single maverick* can jump a problem's complexity from P back up to NP-completeness!

On the other hand, we'll also explore cases where manipulative-action problems remain polynomial-time solvable even if the electorate has a few mavericks. Each such result is, of course, stronger than the analogous polynomial-time claim for the single-peaked case. Such results are typically proven by showing how we can efficiently handle the chaos added by mavericks.

Anna: Where did the term "maverick" come from?

Belle: Samuel Maverick, a colorful Texan who lived from 1803 to 1870, refused to brand his cattle. You can see one of his unbranded cows in Figure 1.10. Unbranded cattle came to be called "mavericks," and the term "maverick" also came to be applied to anyone who is individualistic and unorthodox.

Faliszewski, Hemaspaandra, and Hemaspaandra [15] defined many notions of nearness to single-peakedness, and studied the complexity-theoretic behavior of control and manipulation for electorates of the given nearness types. In this section, for clarity, we will limit ourselves to a sampling of results about two of the more attractive nearness notions that they studied. (Additional nearness measures and discussion of how hard it is to evaluate nearness to single-peakedness of an electorate can be found in [8, 12, 15], and for the complexity of consistency testing and axis-production for the (pure) single-peaked case see [1, 5, 9, 13, 20] and the discussion in Section 2.2 of [17] and

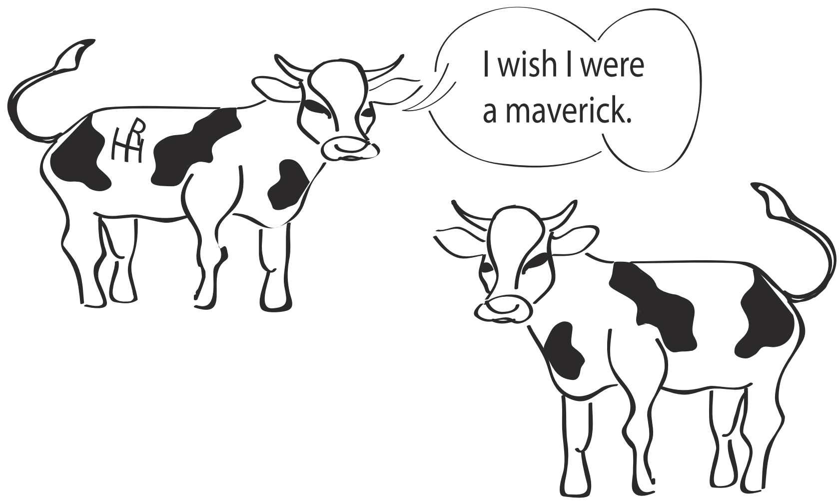


Fig. 1.10 A ruminating cow (left) and a maverick (right)

Section 4 of [15]. Briefly put, although for the single-peaked case this is easy, for "nearly single-peaked" cases such issues can become hard.)

1.5.1 K-Maverick-Single-Peakedness

The first of the two nearness-to-single-peakedness notions that we will study is the notion of a *k-maverick-SP* (for *"k-maverick-single-peaked"*) electorate. As always, our problems will come with a societal axis, L, as part of the input. And a collection of votes is said to be a *k-maverick-SP* electorate if all but at most k of those votes are consistent with the societal axis. When studying manipulative actions problems for the k-maverick-SP case, we require both the input and the after-the-manipulative-action state to be k-maverick-SP electorates. So one cannot in a manipulation problem make so many manipulators be maverick-like that the total number is greater than k. For control-by-adding-voters problems, the entire collection of input votes—both the registered voters and the unregistered ones, viewed together as one big collection—must be a k-maverick electorate.

The motivation for looking at nearly single-peaked electorates is quite compelling. Often electorates are very heavily focused on some issue, such as the sweetness of pumpkin pie in a pie contest, or the degree to which candidates for political office want to redistribute wealth. However, as we saw in the case of Helena, it is perhaps too much to hope that every single person in any reasonably large society will have preferences that mesh with that axis. Helena, due to a medical condition, cared not about sweetness but about the ingredients in the crust. In political elections, even if it seems there is a single clear, salient axis/issue, it is possible that at least a few voters may refuse to vote for the candidate whose position on the axis's issue best matches the voter's, perhaps because the voter has biases, such as refusing to vote for any candidate of a certain religion. Or perhaps the voter perceives differently than others the positions of the candidates on the society's most salient issue. Or perhaps the voter simply doesn't see as the most important issue the same issue that almost everyone in the society sees as their vote-controlling issue. The truth is, although single-peakedness is a very natural notion, probably the right claim to make about it is that in many real-world settings electorates are quite close to being single-peaked. Perfect and pure single-peakedness is too much to hope for in the chaos, confusion, and noise of the real world.

1.5.2 Swoon-Single-Peakedness

The second model of nearness to single-peakedness that we'll study is the notion of a *swoon-SP* (for *"swoon-single-peaked"*) electorate. In that we require that, for each voter, if one removes the top choice of that voter from that voter's ballot and also from the societal axis L, then the resulting ballot is consistent with the resulting axis. This models the case where each voter is perfectly single-peaked along the societal axis, except the voter's top choice may be determined not due to the axis but because the voter has some perhaps emotional, irrational reason to "swoon" for that person. Of course, the model merely allows voters to swoon—it does not force all voters to swoon.

For example, perhaps almost everybody agrees that taxes are the most important issue facing the country. Yet the swoon-SP model would allow some (or all) voters to cast votes where all but the voter's top candidates were ordered in a single-peaked-like fashion, except the top spot in the voter's vote went, for example, to Scarlett Johansson or Arnold Schwarzenegger (Figure 1.11) for some unfathomable swoon-related reason.

Let us now stop swooning and get back to the most salient current issue: learning about the complexity of manipulative actions in nearly single-peaked electorates.

Earlier in this chapter, a remarkably involved dichotomy condition was given, as Theorem 1.5, telling which scoring protocols had their constructive coalitional weighted manipulation complexity in P and for which that problem was NP-complete. For the special case of 3-candidate elections, that theorem simplifies to the following result, which was obtained earlier than Theorem 1.5.

Theorem 1.9 (Faliszewski et al. [17]). *For each scoring protocol* $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$*, the constructive coalitional weighted manipulation problem for the single-peaked case is* NP*-complete if* $\alpha_1 - \alpha_3 > 2(\alpha_2 - \alpha_3) > 0$*, and is in* P *otherwise.*

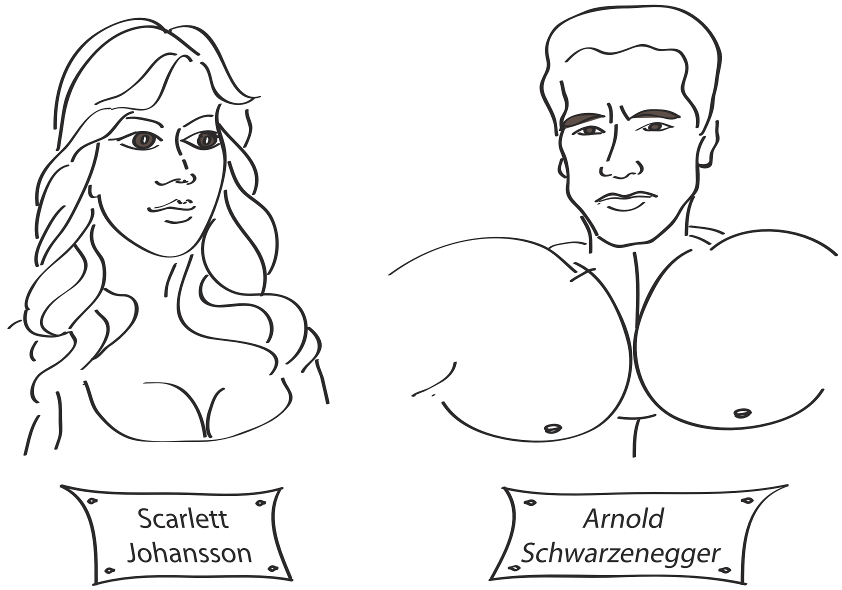


Fig. 1.11 Beneficiaries of swooning

In contrast, for the case of 1-maverick-SP societies, we have the following dichotomy theorem.

Theorem 1.10 (Faliszewski, Hemaspaandra, & Hemaspaandra [15]). *For each scoring protocol* $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$*, the constructive coalitional weighted manipulation problem for 1-maverick-SP societies is* NP*-complete if* $\alpha_2 > \alpha_3$*, and is in* P *otherwise.*

Note that this characterization is quite different than the single-peaked case's characterization. For example, veto elections were NP-complete in the general case, dropped to P in the single-peaked case, but are NP-complete in the 1-maverick-SP case. In fact, since the characterization condition of Theorem 1.10 is identical to the characterization condition of Theorem 1.4, *every* 3-candidate scoring-protocol case that dropped from NP-completeness to P due to single-peakedness is restored to NP-completeness for the case of 1-maverick-SP societies. Even a *single* maverick can result in a tremendous difference in this problem's complexity!

This type of behavior is not limited to the case of just three candidates, or just to the case of 1-maverick-SP. For example, keeping our focus still on veto elections, the following was shown by Faliszewski, Hemaspaandra, and Hemaspaandra [15].

Theorem 1.11 (Faliszewski, Hemaspaandra, & Hemaspaandra [15]). *Let* $m \geq 0$*. For all* $k \geq 0$*, the constructive coalitional weighted manipulation* *problem for* $(m+3)$*-candidate veto elections for* k*-maverick-SP societies is* NP*-complete if* $k > m$*, and is in* P *otherwise.*

For example, for 10-maverick-SP societies, the constructive coalitional weighted manipulation problem for m-candidate veto is in P for $m \in \{0, 1, 2, 13, 14, 15, \ldots\}$ and is NP-complete for $m \in \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. In contrast, veto is in P for any number of candidates for single-peaked societies, and is NP-complete for any number of candidates in the general case. (This behavior can be seen at smaller candidate cardinalities too. For 3-candidate elections in 1-maverick-SP societies the constructive coalitional weighted manipulation problem is NP-complete, but for 4-candidate elections in 1-maverick-SP societies the constructive coalitional weighted manipulation problem is in P.)

Again, we are seeing a type of behavior that is highly unexpected, namely, we are seeing the complexity *drop* as the number of candidates increases from 12 to 13. However, fortified with what we learned from Theorem 1.6, we no longer view this type of behavior as inherently precluded or impossible, and so we don't need to be *too* surprised by it.

That is a good thing, because the swoon-SP case of veto shows the same type of behavior. We have NP-completeness for the 4-candidate case but polynomial-time algorithms for the 5-candidate case.

Theorem 1.12 (Faliszewski, Hemaspaandra, & Hemaspaandra [15]). *For the swoon-SP case, the constructive coalitional weighted manipulation problem for* m*-candidate veto is* NP*-complete for* $m \in \{3, 4\}$*, and is in* P *for* $m \geq 5$*.*

(For $m < 3$, every set of votes is swoon-SP with respect to any societal axis L, so those cases are in fact the general case in disguise.)

So far in this section we have been looking only at manipulation. Does control also show interesting, varied behavior for the nearly single-peaked case? The answer is yes. For brevity, we will give a sense of what results hold not by stating a number of theorems, but rather by giving Table 1.1, which is a restriction to the cases we are interested in of a table from Faliszewski, Hemaspaandra, and Hemaspaandra [15].

Although we won't prove the results of this table, let us briefly mention a key idea behind how one shows that a constant number of mavericks can be tolerated (and for some cases one can even extend this to a logarithmic number of mavericks, see [15]). That idea is from Faliszewski, Hemaspaandra, and Hemaspaandra [15], and they call it *"demaverickification."* Basically, this process takes every maverick voter and shatters him or her into a collection of votes, each of which votes for precisely one of the candidates that that voter approved of (and note that these new votes are, crucially, not maverick votes). Of course, there may be a number of mavericks in the pool of additional votes, and we must decide which ones, if any, to add. When one wraps that up together with the demaverickification process, one ends up with a so-called polynomial-time disjunctive truth-table reduction (as was defined on

Table 1.1 Control complexity results comparison table (adapted from [15]) between two types of nearly single-peaked electorates: the maverick-free single-peaked case and the general case. The "t-approval" column holds for each $t \geq 2$ unless otherwise noted; $t = 1$ is the "plurality" column. N/A means "not applicable" and NPC means "NP-complete."

Control problem	Complexity results for			References
	plurality	t-approval	approval	
CCAC and CCDC				
general case	NPC	NPC	P	[2, 10, 24, 23]
single-peaked	P	P	P	[17, 15]
k-maverick-SP	P for each fixed k	P for each fixed k	P	[15]
swoon-SP	NPC	NPC	N/A	[15]
CCAV				
general case	P	P for $t < 4$ and NPC for $t \geq 4$	NPC	[2, 24, 23]
single-peaked	P	P	P	[2, 17]
k-maverick-SP	P	P for each fixed k	P for each fixed k	[2, 15]
swoon-SP	P	P	N/A	[2, 15]
CCDV				
general case	P	P for $t < 3$, and NPC for $t \geq 3$	NPC	[2, 24, 23]
single-peaked	P	P	P	[2, 17]
k-maverick-SP	P	P for each fixed k	P for each fixed k	[2, 15]
swoon-SP	P	P for $t < 3$ and "2-approximable" for $t \geq 3$	N/A	[2, 24, 15]

page **??**) from the k-maverick-SP case to the single-peaked case. That is, we can turn one k-maverick-SP instance into a large—but not too large—number of maverick-free single-peaked instances, such that the original problem has the answer "yes" if and only if at least one of our new instances has the answer "yes." This in some sense lets the k-maverick-SP case ride on the coattails of the (maverick-free) single-peaked case, although with quite a bit of work being done to make this possible. However, the bottom line is that this is enough to cast many problems into P for the k-maverick-SP case.

"This has been fun," says Belle to her friends, "but I think we now deserve an extra treat. Let's all go and have a yummy meal of 'pumpkin-pie surprise,' which of course is pumpkin pie surprisingly topped with *more* pumpkin pie!" Belle's friends' eyes open wide with delight, and—now with a better understanding of the complexity of manipulative actions in single-peaked elections—they all happily race off to eat.

References

1. Bartholdi III, J., Trick, M.: Stable matching with preferences derived from a psychological model. Operations Research Letters **5**(4), 165–169 (1986)
2. Bartholdi III, J., Tovey, C., Trick, M.: How hard is it to control an election? Mathematical and Computer Modelling **16**(8/9), 27–40 (1992)
3. Black, D.: On the rationale of group decision-making. Journal of Political Economy **56**(1), 23–34 (1948)
4. Black, D.: The Theory of Committees and Elections. Cambridge University Press (1958)
5. Booth, K.: Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems. SIAM Journal on Computing **7**(3), 273–279 (1978)
6. Booth, K., Lueker, G.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. Journal of Computer and System Sciences **13**(3), 335–379 (1976)
7. Brandt, F., Brill, M., Hemaspaandra, E., Hemaspaandra, L.: Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 715–722. AAAI Press (2010)
8. Bredereck, R., Chen, J., Woeginger, G.: Are there any nicely structured preference profiles nearby? In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 62–68. AAAI Press/IJCAI (2013)
9. Doignon, J., Falmagne, J.: A polynomial time algorithm for unidimensional unfolding representations. Journal of Algorithms **16**(2), 218–233 (1994)
10. Elkind, E., Faliszewski, P., Slinko, A.: Cloning in elections: Finding the possible winners. Journal of Artificial Intelligence Research **42**, 529–573 (2011)
11. Elkind, E., Lackner, M., Peters, D.: Preference restrictions in computational social choice: A survey. Tech. Rep. arXiv:2205.09092v1 [cs.GT], ACM Computing Research Repository (CoRR) (2022)
12. Erdélyi, G., Lackner, M., Pfandler, A.: Computational aspects of nearly single-peaked electorates. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, pp. 283–289. AAAI Press (2013)
13. Escoffier, B., Lang, J., Öztürk, M.: Single-peaked consistency and its complexity. In: Proceedings of the 18th European Conference on Artificial Intelligence, pp. 366–370. IOS Press (2008)
14. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: How hard is bribery in elections? Journal of Artificial Intelligence Research **35**, 485–532 (2009)
15. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: The complexity of manipulative attacks in nearly single-peaked electorates. Artificial Intelligence **207**, 69–99 (2014)

16. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Llull and Copeland voting computationally resist bribery and constructive control. Journal of Artificial Intelligence Research **35**, 275–341 (2009)
17. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. Information and Computation **209**(2), 89–107 (2011)
18. Fishburn, P.: Condorcet social choice functions. SIAM Journal on Applied Mathematics **33**(3), 469–489 (1977)
19. Fitzsimmons, Z., Lackner, M.: Incomplete preferences in single-peaked electorates. Journal of Artificial Intelligence Research **67**, 797–833 (2020)
20. Fulkerson, D., Gross, G.: Incidence matrices and interval graphs. Pacific Journal of Mathematics **15**(5), 835–855 (1965)
21. Gailmard, S., Patty, J., Penn, E.: Arrow's theorem on single-peaked domains. In: E. Aragonès, C. Beviá, H. Llavador, N. Schofield (eds.) The Political Economy of Democracy, pp. 335–342. Fundación BBVA (2009)
22. Hemaspaandra, E., Hemaspaandra, L.: Dichotomy for voting systems. Journal of Computer and System Sciences **73**(1), 73–83 (2007)
23. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Anyone but him: The complexity of precluding an alternative. Artificial Intelligence **171**(5–6), 255–285 (2007)
24. Lin, A.: Solving hard problems in election systems. Ph.D. thesis, Rochester Institute of Technology, Rochester, NY, USA (2012)
25. Walsh, T.: Uncertainty in preference elicitation and aggregation. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 3–8. AAAI Press (2007)