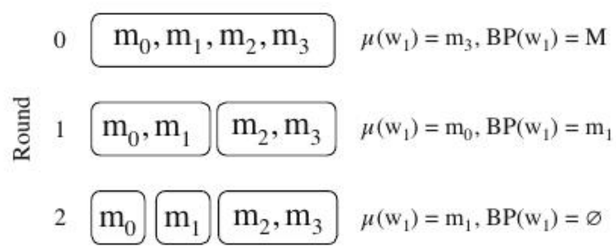

Figure 1: Evolution of preferences for w_1 in Example 1 using RBH.

reduce $MR(\mu)$; see line 4 of Alg. 2. This can be viewed as a form of *current solution* elicitation [Boutilier *et al.*, 2006; Braziunas and Boutilier, 2007; Lu and Boutilier, 2011b], where queries are driven by the regret-optimal solution.

The selection of individuals for querying occurs in lines 4–10. Intuitively, querying the regret-inducing individuals q and their blocking partners has the greatest potential to lower the max regret of the current matching. Lines 4–6 ask q to split the block containing both its current partner $\mu(q)$ and blocking $r \in BP(q)$ when they are in the same block. This will either confirm $\mu(q)$ or r as preferred by q, or will reduce their PMR by a half. If q's blocking partner r is not in the same block as $\mu(q)$, we instead query r, asking him to split the block containing q and $\mu(r)$ if they are in the same block for similar reasons (lines 8–10): note that r's PMR for q is at least as great as that of q for r. In both cases, queries are targeted toward individuals where a single block contains the "relevant" uncertainty required to identify the optimal "attainable" partner, in some loose sense simulating binary search. If neither of these conditions holds for any $q \in RI(\mu)$, we resort to asking each individual to split their largest block to ensure that elicitation does not stall.

Cognitive Costs. Asking a person to "split a block" based on preferences is a very different form of query than those asked by GS. Furthermore, not all splitting queries are equally difficult, since the sizes of the blocks that are split varies (and may contain potential partners that are either very close of very distant from each other in terms of preference). To meaningfully compare RBH with GS, we measure the number and cognitive difficulty of the pairwise comparisons required to answer their queries. We show that, in practice, RBH outperforms GS with respect to the total *cognitive complexity* of queries, as well as the number of queries itself.

One natural measure for comparing how difficult queries are would be to simply consider the number of pairwise comparisons required. Using a Quickselect-style algorithm [Hoare, 1961], a block of size z can be split using $O(z)$ comparisons (assuming reasonable pivot choices). For GS, proposers can select the next best partner to propose to using $O(z)$ comparisons when selecting from z unproposed candidates; and acceptors similarly must make comparisons linear in the number of received proposals. Of course, not all pairwise comparisons are equally difficult. Intuitively, comparing two partners widely separated in one's ranking is easier than comparing two that are close, a fact reflected in many psychometric and behavioral economics models of choice [Louviere *et al.*, 2000; Camerer *et al.*, 2003]. To reflect this, we measure the difficulty of a comparison using the *Luce-Shepard* choice model [Luce, 1959; Shepard, 1959], in which the probability of choosing a lower ranked item over a higher ranked item decreases exponentially with their *separation* or difference in utility. Equating degree of difficulty with choice error, given an underlying ranking $\succ_q$ for person q, temperature $\gamma \geq 0$, and threshold τ, the cost of comparing r with r' is:

$$c(r, r') = e^{\gamma(n - \min(|s_q(r, \succ_r) - s_q(r', \succ_r)|, \tau))} \quad (7)$$

(Note: $\gamma = 0$ makes all comparisons equally difficult.) We assess the cognitive complexity of both RBH and GS below.

Table 1: Average number of queries (std. dev.) until max regret 0, Mallows models: $n = 20$, 30 trials.

4.2 Empirical Results

We now describe experiments that test the effectiveness of our elicitation schemes. We draw preference profiles from specific preference distributions, and measure the average number of queries and rounds needed to reach a stable matching (i.e., max regret zero), the cognitive complexity of those queries, and their anytime performance w.r.t. *approximate stability*. We also assess their computational performance.

Number of queries. We first consider small instances with $n = 20$ (i.e., 20 men, 20 women) to allow use of the IP (Alg. 1) to compute true minimax regret. Table 1 shows the average number of queries (over 30 trials) per proposer/acceptor and number of rounds required by different schemes using a Mallows models to generate preferences with $\phi = 1$ (impartial culture) and $\phi = 0.2$ (strongly correlated preferences). GS marginally outperforms all RBH schemes when preferences are completely uncorrelated, but except for IP, the differences are not statistically significant for proposers, and the advantage for acceptors is quite small. With strongly correlated preferences, the RBH schemes (apart from IP) reach a stable matching with *far fewer* queries per proposer, and differ from GS by less than one query per acceptor. In all cases, the RBH schemes (excluding IP) ask fewer than $\log n$ queries per person; thus they even outperform binary search for a *known* target partner (of course, the target is not known *a priori*). The IP scheme works surprisingly poor. This is, in large part, due to the fact that the matchings computed by the IP often do not match individuals to partners in the partitions that they ultimately end up being matched in.

For these small instances, the IP requires roughly 0.5 sec. to compute the MMR matching (and hence generate queries in a given round), but becomes impractical beyond $n = 30$. We do not consider IP further on the larger instances below. The other RBH schemes require 0.01–0.06 sec. on average to generate a matching, compute its max regret, and generate queries. GS is much faster, but does not compute max regret; thus it offers no anytime guarantees (as we discuss below).