JOURNAL OF ALGORITHMS **20,** 431–442 (1996)
ARTICLE NO. 0020

An Exact Analysis of Stable Allocation

Donald E. Knuth

Computer Science Department, Stanford University, Stanford, California 94305-2140

Received September 21, 1994; revised March 20, 1995

Shapley and Scarf (*J. Math. Econ.* **1** (1974), 23–38) introduced a notion of stable allocation between traders and indivisible goods, when each trader has rank-ordered each of the goods. The purpose of this note is to prove that the distribution of ranks after allocation is the same as the distribution of search distances in uniform hashing, when the rank-orderings are independent and uniformly random. Therefore, the average sum of final ranks is just $(n + 1)H_n - n$, and the standard deviation is $O(n)$. The proof involves a family of interesting one-to-one correspondences between permutations of a special kind.

INTRODUCTION

Suppose n traders have n indivisible goods to trade, and each trader k has ranked the goods of all traders (including himself) as a permutation

$$p_k = p_{k1} p_{k2} \dots p_{kn}$$

of $\{1, 2, \dots, n\}$. If i precedes j in this list, we write $i > j\ (k)$ and say that k prefers i to j. An allocation of goods to traders is a permutation $g_1 \dots g_n$ of $\{1, \dots, n\}$ such that trader k gets g_k. Shapley and Scarf [8] defined what they called a "core allocation" g, which is *stable* in the following sense: If C is any coalition of traders (a nonempty subset), and if h is any allocation of the goods of C to the members of C, then

$$h_k \geq g_k\ (k) \text{ for all } k \in C \quad \Rightarrow \quad h_k = g_k \text{ for all } k \in C. \qquad (*)$$

For example, suppose $n = 3$ and the preference rankings are

$$p_1 = 2\ 1\ 3,$$
$$p_2 = 3\ 1\ 2,$$
$$p_3 = 2\ 3\ 1.$$

0196-6774/96 $18.00

Then just three of the six possible allocations satisfy $(*)$ when C is the full set $\{1, 2, 3\}$, namely 2 1 3, 2 3 1, and 1 3 2. But 2 1 3 is unstable because traders 2 and 3 can both improve their lot by swapping goods between themselves. Similarly, 2 3 1 is unstable, but it is not quite as bad: If the coalition $\{2, 3\}$ exchanges goods, trader 3 is happier than he was before, while 2 is no worse off. The remaining allocation, 1 3 2, is stable.

It is not immediately obvious that a stable allocation always exists, for all $n!^n$ possible ranking sequences. But Shapley and Scarf presented an algorithm by Gale that always finds one. In fact, there is always *exactly one* stable allocation. Gale's procedure is similar to the famous Gale–Shapley algorithm for stable marriage [2, 4], but it incorporates a new twist.

Frieze and Pittel [1] recently analyzed Gale's algorithm and discovered some remarkable simplifications in the course of their study. One of the main purposes of the present note is to exhibit some underlying combinatorial structure that accounts for the surprising phenomena they discovered. Frieze and Pittel proved, among other things, that the total sum of ranks in the stable allocation g (i.e., the sum $r_1 + \cdots + r_n$, where $g_k = p_{kr_k}$) is between $(\frac{1}{2} - \epsilon)n \ln n$ and $(1 + \epsilon)n \ln n$ with high probability as $n \to \infty$, assuming that the preferences p_k are independent and uniformly random. We will deduce the exact distribution of $r_1 + \cdots + r_n$, showing in particular that its mean value is $(n + 1)H_n - n$, and in fact we will see that the joint distribution of the multiset $\{r_1, \dots, r_n\}$ has a particularly simple form.

1. UNIFORM HASHING

First let us consider a simpler problem, namely, to find an allocation g that satisfies $(*)$ when C is the full set $\{1, \dots, n\}$ but not necessarily for any other C. Such an allocation is *locally optimal*, in the sense that no trader can improve his selection unless some other trader loses ground. It is easy to achieve such an allocation by simply letting g_k be the first item of list p_k that is not in $\{g_1, \dots, g_{k-1}\}$, for $k = 1, \dots, n$.

This trivial allocation algorithm, "first-come, first-served," is precisely the method of *uniform hashing* that arises in the study of information retrieval [5], when the preference lists (called "hash sequences" in that context) are randomly chosen. The analysis of uniform hashing is particularly simple, and we will see below that stable allocation can be reduced to the same analysis.

A more general way to obtain a locally optimal allocation is to let π be any permutation of $\{1, \dots, n\}$, and then to let $g_{\pi(k)}$ be the first item of $p_{\pi(k)}$ that is not in $\{g_{\pi(1)}, \dots, g_{\pi(k-1)}\}$, for $k = 1, \dots, n$. The permutation π give top priority to trader $\pi(1)$, then to $\pi(2)$, and so on. Indeed, *every* allocation that satisfies $(*)$ for $C = \{1, \dots, n\}$ will be found by this method, for some π. The reason is that we must have $g_k = p_{k1}$ for some k, in any locally optimum g, because the mapping from k to p_{k1} for all k always contains a cycle; any such cycle can be used to improve an allocation in which no trader has his first choice. Let $\pi(1)$ be any value of k with $g_k = p_{k1}$. Remove k from all preference lists and apply the same reasoning recursively to the remaining $n-1$ traders. This defines a permutation $\pi(1)\pi(2)\ldots\pi(n)$ such that $g_{\pi(k)}$ is the favorite of trader $\pi(k)$ in $\{1,\ldots,n\} \setminus \{g_{\pi(1)},\ldots,g_{\pi(k-1)}\}$.

Since every locally optimum allocation is obtained in this way using some π, the stable allocation must itself result from some π. And when the preference lists are random, any π behaves like any other. Thus, we might expect that stable allocation statistics are essentially identical to the statistics of uniform hashing. This, in fact, is true, but we must be careful to make the argument rigorous.

2. AN ALGORITHM

Let us now consider a simple algorithm that computes the stable allocation, given any sequence of preference rankings $p_1 \ldots p_n$. The following procedure is a sequential variant of Gale's parallel method, analogous to the McVitie–Wilson version [7] of Gale and Shapley's original stable marriage algorithm. The basic idea is to look for cycles among the traders' best choices, and to put such cycles into the allocation whenever they are found.

A1. [Initialize.] Set $(q_1,\ldots,q_n) \leftarrow (-1,\ldots,-1)$, $(r_1,\ldots,r_n) \leftarrow (0,\ldots,0)$, $(g_1,\ldots,g_n) \leftarrow (0,\ldots,0)$, and $t \leftarrow 0$. (During this algorithm, t will be a trader who makes proposals to other traders, or zero when a new trader needs to enter the picture. Variable q_k will be the number of a trader who currently wants traders k's goods, or $q_k = 0$ if trader k has expressed interest in somebody else's wares but nobody has reciprocated; $q_k = -1$ if trader k has not yet entered. Variable r_k is the position of trader k in his list, the number of proposals he has made. Variable g_k is trader k's allocation, or 0 if no allocation has yet been made.)

A2. [Introduce a new trader.] (At this point $t = 0$, and $g_k = 0$ iff $q_k = -1$. The traders with $g_k > 0$ have been assigned a permutation of their goods.) If all g_k are nonzero, the algorithm terminates. Otherwise, set t to some k with $g_k = 0$, and set $q_t \leftarrow 0$.

A3. [Propose.] Increase r_t by 1, then set $s = p_{tr_t}$. If $g_s > 0$, repeat this step. (Trader t has expressed interest in his best remaining choice, s.)

A4. [Is s spoken for?] If $q_s \geq 0$, go to step A5. Otherwise set $q_s \leftarrow t$ and $t \leftarrow s$, then return to A3.

A5. [Remove a cycle.] (There is now a cycle $s = s_1 \to s_2 \to \cdots \to s_m = s$, where s_{j+1} is the best remaining choice of s_j. This cycle must be part of any stable allocation, so we incorporate it into g.) Set $t \leftarrow q_s$; then repeatedly set $g_s \leftarrow p_{sr_s}$ and $s \leftarrow p_{sr_s}$ until finding $g_s > 0$. If $t = 0$, return to A2; otherwise, go to A3.

In step A3 there is always a path

$$0 = t_0 \to t_1 \to t_2 \to \cdots \to t_m = t$$

connecting all traders k such that $g_k = 0$ and $q_k \geq 0$. Trader t_1 entered in step A2, and t_{j+1} is the best remaining choice of t_j, for $1 \leq j < m$; also $q_{t_j} = t_{j-1}$ for $1 \leq j \leq m$. These invariant relations justify the parenthesized assertions within the algorithm.

The final allocation $g_1 \ldots g_n$ is stable. For if $h_k \leq g_k\ (k)$ for h_k in some coalition C, and if some $h_k < g_k\ (k)$, then $h_k = p_{kr}$ for some $r < r_k$, so h_k was rejected by the algorithm. When the algorithm changed r_k from r to $r+1$, it had already found h_k to be the best remaining choice of some other trader s, and it had assigned $g_s = h_k$. Therefore, $h_s > g_s\ (s)$.

Moreover, the stable allocation is unique. If g is assigned differently on any cycle that leads to step A5, that cycle will be a coalition violating (2).

3. A CONSTRUCTIVE LEMMA

We have observed that the stable allocation will be found by a first-come-first-served algorithm equivalent to uniform hashing, using at least one permutation π to give priority to the traders. For example, in the Introduction we considered a case where $n = 3$ and the stable allocation was 1 3 2. Any π in which trader 3 has priority over trader 1 will find this allocation.

We can also consider permutations of the preference lists. Let $\sigma = \sigma(1) \ldots \sigma(n)$ be a permutation of $\{1, \ldots, n\}$, and suppose that trader $\sigma(k)$ uses list $p'_{\sigma(k)} = p_k$. This will permute the locally optimum allocations, and it may also change the stable allocation. For example, if σ is 2 1 3, so that

$$p'_2 = 2\ 1\ 3,$$

$$p'_1 = 3\ 1\ 2,$$

$$p'_3 = 2\ 3\ 1,$$

the stable allocation becomes $g'_2 g'_1 g'_3 = 2\ 1\ 3$, because g'_2 must be 2 and then g'_3 must be 3. Allocating goods in order 2 1 3 was the worst of the locally stable alternatives when σ was the identity permutation 1 2 3, but it is best in the modified problem. Shuffling the preference lists corresponds to shuffling the goods that the traders started with.

We are now ready to prove a key fact about stable allocation. Let us say that the prioritization π is *consistent with* the shuffling σ, with respect to preferences $p = p_1 \ldots p_n$, if the locally optimum allocation $g_1 \ldots g_n$ obtained by uniform hashing with priorities π is the stable allocation $g'_{\sigma(1)} \ldots g'_{\sigma(n)}$ when $p'_{\sigma(k)} = p_k$. For example, $\pi(1)\pi(2)\pi(3) = 1\ 3\ 2$ produces the locally optimum $2\ 1\ 3$, so π is consistent with the shuffling $\sigma(1)\sigma(2)\sigma(3) = 2\ 1\ 3$ just considered.

LEMMA. *Let p be any sequence of preference lists. There is a one-to-one correspondence between all permutations π of $\{1, \ldots, n\}$ and all permutations σ such that, if π corresponds to σ, the prioritization π is consistent with the shuffling σ.*

Proof. Given p and π, suppose uniform hashing with priorities π produces the locally optimum allocation $g_1 \ldots g_n$. Write the preference lists in rows, with each g_k circled in its list p_k. Delete all elements to the right of g_k.

We will construct a shuffling σ whose stable allocation agrees with g. The construction involves two dynamically growing sets X and Y, whose significance will become clear momentarily. Initially, $X \leftarrow \emptyset$ and Y is the set of all k where $g_k = p_{k1}$ (i.e., all row numbers in which the circled element is all by itself). Set $m \leftarrow 0$; as the construction proceeds, we will have defined $\sigma(\pi(1)), \ldots, \sigma(\pi(m))$ as a permutation of $\{g_{\pi(1)}, \ldots, g_{\pi(m)}\}$, and we will have $X \subseteq Y$, $\pi(m+1) \in Y \setminus X$.

Find the minimum k in $m < k \leq n$ such that either $k = n$ or ($\pi(k+1) \in Y \setminus X$ and $\pi(k+1) > \pi(m+1)$) or $\pi(k+1) \notin Y$. Define $\sigma(\pi(j)) = g_{\pi(j+1)}$ for $m < j < k$ and $\sigma(\pi(k)) = g_{\pi(m+1)}$. If $k = n$, the construction is complete. Otherwise, remove $g_{\pi(m+1)}, \ldots, g_{\pi(k)}$ from all preference lists where they are not circled. If $\pi(k+1) \notin Y$, set $X \leftarrow Y$ and let Y be the set of all rows whose first elements are now circled. (Since π defines g by uniform hashing, $\pi(k+1)$ will be in the new Y.) Set $m \leftarrow k$ and repeat the instructions of this paragraph.

A worked example will help clarify this construction. Let

$$\pi(1) \ldots \pi(9) = 5\ 3\ 4\ 9\ 1\ 8\ 2\ 7\ 6.$$

Table 1 shows a sequence of preference lists for $n = 9$ in which π defines the locally optimum allocation indicated by circled elements. All elements to the right of those circled have been erased, since they are irrelevant for our present purposes. The stable allocation determined by these preference lists happens to coincide with the circled elements in Table 1, so in this case the priorities π produce the stable allocation; but our construction works for any π, whether or not its locally optimum allocation is stable. Initially, $m = 0$, $X = \emptyset$, and $Y = \{1, 2, 3, 5, 6, 9\}$. According to the

TABLE 1
Preference Lists and Their Stable Allocation

1:	③	4:	1 ⑤	7:	5 3 ⑧
2:	④	5:	⑨	8:	9 ⑦
3:	①	6:	⑥	9:	②

rules stated, we proceed to set $k = 2$, since $\pi(3) \notin Y$. So we define $\sigma(5) = g_3 = 1$ and $\sigma(3) = 9$; then we delete 1 and 9 from lists 4 and 8, and we set $X \leftarrow \{1, 2, 3, 5, 6, 9\}$, $Y \leftarrow \{1, 2, 3, 4, 5, 6, 8, 9\}$, $m \leftarrow 2$. Next, $k = 5$ since $\pi(6) = 8 \in Y \setminus X$ and $8 > 4 = \pi(3)$. This time $\sigma(4) = 2$, $\sigma(9) = 3$, $\sigma(1) = 5$. We delete 2, 3, and 5 where they are not circled. After setting $m \leftarrow 5$ we have $k = 7$, because $\pi(8) \notin Y$. (Note that $7 \notin Y$, even though row 7 now contains only its circled element. The construction changes Y only in the case $\pi(k + 1) \notin Y$.) This time $\sigma(8) = 4$, $\sigma(2) = 7$, $X \leftarrow \{1, 2, 3, 4, 5, 6, 8, 9\}$, $Y \leftarrow \{1, \ldots, 9\}$, $m \leftarrow 7$. On the final round we set $\sigma(7) = 6$ and $\sigma(6) = 8$; the shuffled preference lists are shown in Table 2. It is easy to verify that their stable allocation matches that of Table 1, using the algorithm given earlier.

The inverse construction is analogous. If σ is any shuffling, circle its stable allocation and prepare an array like Table 2. Begin with $m \leftarrow 0$, $X = \emptyset$, and Y as before. Then repeatedly consider all cycles

$$\sigma(a_1) \leftarrow \sigma(a_2) \leftarrow \cdots \leftarrow \sigma(a_t) \leftarrow \sigma(a_1)$$

formed by elements $\{\alpha_1, \ldots, a_t\} \subseteq Y \setminus \{\pi(1), \ldots, \pi(m)\}$. Here $\sigma(a) \leftarrow \sigma(b)$ means that $\sigma(a)$ is the (circled) element in list $\sigma(b)$; for example, $\sigma(4) \leftarrow \sigma(9)$ in Table 2 because the circled element in list $\sigma(9) = 3$ is $2 = \sigma(4)$. The properties of stable allocation guarantee that at least one such cycle exists, and our inverse construction will guarantee that each cycle will contain at least one $a_i \notin X$. Call the largest such a_i the *cycle leader* and renumber the subscripts so that a_1 is the cycle leader. Take the

TABLE 2
Shuffled Precedence Lists Having the Same Stable Allocation as Table 1

5:	③	2:	1 ⑤	6:	5 3 ⑧
7:	④	1:	⑨	4:	9 ⑦
9:	①	8:	⑥	3:	②

cycle with smallest leader, and set $\pi(m+1) \leftarrow a_1, \ldots, \pi(m+t) \leftarrow a_t$. Remove $\sigma(a_1), \ldots, \sigma(a_t)$ from the tableau in places where they are not circled. Then set $m \leftarrow m + t$ and repeat the same process until all cycles have been recorded in π. Then set $X \leftarrow Y$ and let Y be the row numbers that now have but a single element. Repetition of these steps will produce a priority permutation π consistent with σ.

It is not difficult to verify that these constructions invert each other. The reader will find easily, for example, that the permutation $\sigma(1) \ldots \sigma(9) = 5\ 7\ 9\ 2\ 1\ 8\ 6\ 4\ 3$ in Table 2 leads back to $\pi(1) \ldots \pi(9) = 5\ 3\ 4\ 9\ 1\ 8\ 2\ 7\ 6$. ∎

4. A THEOREM

The lemma we have just proved makes it easy to establish the main result of this note. We say that uniform hashing on $p_1 \ldots p_n$ with priorities π produces ranks $r_1 \ldots r_n$ if $r_{\pi(k)}$ is minimum such that

$$p_{\pi(k) r_{\pi(k)}} \notin \{ p_{\pi(1) r_{\pi(1)}}, \ldots, p_{\pi(k-1) r_{\pi(k-1)}} \}$$

for $1 \le k \le n$.

THEOREM. *When preference lists $p_1 \ldots p_n$ are independent and uniformly random, the probability that the stable allocation $g_1 \ldots g_n = p_{1r_1} \ldots p_{nr_n}$ has a given value of the (unordered) multiset $\{r_1, \ldots, r_n\}$ is the same as the probability that uniform hashing yields $\{r_1, \ldots, r_n\}$.*

Proof. Let $\{r_1, \ldots, r_n\}$ be any given multiset. If $p = p_1 \ldots p_n$ is any sequence of preferences and σ is any permutation of $\{1, \ldots, n\}$, let $s(\sigma, p) = 1$ if $\{r_1, \ldots, r_n\}$ is the multiset of ranks in the stable allocation when trader $\sigma(k)$ has the preference list p_k; otherwise, $s(\sigma, p) = 0$. Then the probability that stable allocation on random preferences has ranks $\{r_1, \ldots, r_n\}$ is

$$\frac{1}{n!^n} \sum_p s(\sigma, p)$$

for any fixed σ.

Similarly, if π is any permutation of $\{1, \ldots, n\}$, let $h(\pi, p) = 1$ iff $\{r_1, \ldots, r_n\}$ is the multiset of ranks produced by uniform hashing with priorities π. Then the probability that uniform hashing on random preferences has ranks $\{r_1, \ldots, r_n\}$ is

$$\frac{1}{n!^n} \sum_p h(\pi, p)$$

for any fixed π.

We want to show that these sums are equal. This is now obvious, because the lemma implies that

$$\sum_\sigma \sum_p s(\sigma, p) = \sum_p \sum_\sigma s(\sigma, p) = \sum_p \sum_\pi h(\pi, p) = \sum_\pi \sum_p h(\pi, p)$$

and we simply divide by $n!^{n+1}$. ∎

Note that this proof of the theorem remains valid even when the preference lists $p_1 \ldots p_n$ are not uniformly random. All we are assuming is a symmetry condition that shuffled preference lists $p_{\sigma(1)} \cdots {}_{\sigma(n)}$ have the same distribution for all σ.

5. COROLLARIES

The analysis of uniform hashing is quite simple, so our theorem immediately characterizes many properties of the ranks in random stable allocations. For example, let us find the expected value of

$$(z + r_1)(z + r_2) \ldots (z + r_n);$$

this polynomial is clearly a function of the multiset $\{r_1, \ldots, r_n\}$, so we can analyze it by considering its behavior with respect to uniform hashing.

Let q_{kj} be the probability that $r_k > j$ in uniform hashing. This is the probability that $p_{k1}, \ldots, p_{kj} \in \{p_{1r_1}, \ldots, p_{(k-1)r_{k-1}}\}$, so

$$q_{kj} = \left(\frac{k-1}{n}\right)\left(\frac{k-2}{n-1}\right) \cdots \left(\frac{k-j}{n-j+1}\right) = \binom{k-1}{j} \Big/ \binom{n}{j}. \tag{1}$$

Standard binomial coefficient summation techniques [3] show that

$$\sum_{j=0}^{\infty} \binom{j}{m} q_{kj} = \frac{n+1}{n+m+2-k} \binom{k-1}{m} \Big/ \binom{n+m+1-k}{m}. \tag{2}$$

The expected value of $(z + r_1)\ldots(z + r_n)$ is therefore

$$\sum_{r_1,\ldots,r_n} \prod_{k=1}^{n} (q_{k(r_k-1)} - q_{kr_k})(z + r_k)$$
$$= \prod_{k=1}^{n} \sum_{r=1}^{\infty} (q_{k(r-1)} - q_{kr})(z + r)$$
$$= \prod_{k=1}^{m} \left(z + \sum_{j=0}^{\infty} q_{kj}\right)$$
$$= \prod_{k=1}^{n} \left(z + \frac{n+1}{n+2-k}\right) = \frac{1}{(n+1)!} \prod_{k=2}^{n+1} (kz + n + 1). \quad (3)$$

In particular, the expected value of $r_1 + \cdots + r_n$, which is the coefficient of z^{n-1}, is

$$\sum_{k=1}^{n} \frac{n+1}{n+2-k} = (n+1)(H_{n+1} - 1) = (n+1)H_n - n. \quad (4)$$

The other coefficients can be expressed in terms of Stirling cycle numbers if we note that

$$E(z + r_1)\ldots(z + r_n)(z + n + 1) = \frac{1}{(n+1)!} \prod_{k=1}^{n+1} (kz + n - 1)$$
$$= \sum_{k=0}^{n+1} \left[{n+2 \atop k+1} \right] \frac{(n+1)^k}{(n+1)!} z^{n+1-k}. \quad (5)$$

For example, the coefficient of z^{n-2} in $E(z + r_1)\ldots(z + r_n)$ is

$$\left[{n+2 \atop 3} \right] \frac{(n+1)^2}{(n+1)!} - (n+1)\left[{n+2 \atop 2} \right] \frac{(n+1)}{(n+1)!}$$
$$+ (n+1)^2 \left[{n+2 \atop 1} \right] \frac{1}{(n+1)!}$$
$$= (n+1)^2 \left(\tfrac{1}{2}\left(H_{n+1}^2 - H_{n+1}^2\right) - H_{n+1} + 1 \right)$$
$$= \frac{(n+1)^2}{2} \left(H_n^2 - H_n^{(2)}\right) - n(n+1)(H_n - 1); \quad (6)$$

see [3, exercise 6.33].

So far we have used only the case $m = 0$ of (2). A similar argument, using $m = 1$, shows that

$$E\big(z + r_1^2\big)\dots\big(z + r_n^2\big) = \prod_{k=1}^{n}\left(z + \frac{(n+1)(n+1+k)}{(n+2-k)(n+3-k)}\right). \quad (7)$$

In particular,

$$\begin{aligned} E\big(r_1^2 + \cdots r_n^2\big) &= \sum_{k=1}^{n} \frac{(n+1)(n+1+k)}{(n+2-k)(n+3-k)} \\ &= (n+1)\sum_{k=1}^{n}\left(\frac{2n+4}{(n+2-k)(n+3-k)} - \frac{1}{n+2-k}\right) \\ &= (n+1)(n - H_{n+1} + 1) = (n+1)(n - H_n) + n. \quad (8) \end{aligned}$$

Hence, by (6) and (8),

$$\begin{aligned} E\big((r_1 + \cdots + r_n)^2\big) &= E\big(r_1^2 + \cdots r_n^2\big) + 2[z^{n-2}]E(z + r_1)\dots(z + r_n) \\ &= (n+1)^2\big(H_n^2 - H_n^{(2)}\big) - (n+1)(2n+1)H_n \\ &\quad + n(3n+4). \quad (9) \end{aligned}$$

The expected value of the variance of the ranks is therefore

$$E\left(\frac{r_1^2 + \cdots + r_n^2}{n}\right) - E\left(\left(\frac{r_1 + \cdots + r_n}{n}\right)^2\right) = n + O(\log n)^2 \quad (10)$$

while the variance of the rank sum is

$$\begin{aligned} &E\big((r_1 \cdots + r_n)^2\big) - \big(E(r_1 + \cdots + r_n)\big)^2 \\ &\quad = 2n(n+2) - (n+1)^2 H_n^{(2)} - (n+1)H_n \\ &\quad = \left(2 - \frac{\pi^2}{6}\right)n^2 + O(n \log n). \quad (11) \end{aligned}$$

The final rank r_n in uniform hashing is uniformly distributed in $\{1, \dots, n\}$. Therefore, the probability is $\geq \frac{1}{2}$ that at least one trader in a random stable allocation will have rank $\geq \frac{1}{2}n$. (He will be left with a piece of goods he does not like very much, since it is in the lower half of his list.) Indeed, the probability that $\max(r_1, \dots, r_n) \leq \frac{1}{2}n$ is exactly

$$(1 - q_{1m})(1 - q_{2m})\dots(1 - q_{nm}),$$

where $m = \lfloor \frac{1}{2}n \rfloor$; this is asymptotically

$$\big(1 - \tfrac{1}{2}\big)\big(1 - \tfrac{1}{4}\big)\big(1 - \tfrac{1}{8}\big)\big(1 - \tfrac{1}{16}\big)\dots \approx 0.288788. \quad (12)$$

6. CONCLUSIONS AND CONJECTURES

The running time of the simple algorithm we have presented for stable allocation is essentially proportional to the sum of ranks in the unique allocation, $r_1 + \cdots + r_n$. We have proved that the statistical properties of any symmetric function of $(r_1 \ldots r_n)$ are identical to the corresponding statistics for uniform hashing, provided only that the distribution of preference lists $p_1 \ldots p_n$ is invariant under shuffling. When the preferences are uniformly random, the expected value of $r_1 + \cdots + r_n$ is exactly $(n+1)H_n - n$, and the standard deviation is $O(n)$.

Uniform hashing is equivalent to the classical stable marriage problem when all the girls have the same preference list. (See [6, pp. 65–67].) Perhaps it is worthwhile to repeat here the main research problem about stable marriages that was advertised in the author's lectures of 1975 [6] and not yet resolved: If the girls have any fixed set of preferences and the boys propose at random, is the expected rank rum $r_1 + \cdots + r_n$ of the male-optimum stable marriage always $\geq (n+1)H_n - n$? In other words, does the case of equal preferences for the girls (uniform hashing) give the greatest lower bound for $E(r_1 + \cdots + r_n)$? If so, the average would be tightly bounded, because the upper bound $(n-1)H_n + 1$ is easy to prove [6, p. 43].

In fact, computer experiments for small n suggest that the maximum value of $E(r_1 + \cdots + r_n)$, when the girls have a fixed set of preferences and the boys propose independently at random, is obtained if and only if the girls' preferences are cyclic, in the sense that we could rename boys and girls so that girl j's kth choice is congruent to $j + k \pmod{n}$.

Both conjectures about min and max $E(r_1 + \cdots + r_n)$ have been verified by exhaustive enumeration when $n \leq 4$, and in several hundred random experiments when $n = 5$. Presumably there is a (simple?) way to prove that, in some sense, the more the girls agree in their ranking, the less the men will have to propose, on the average.

Is there a simple expression for $E(r_1 + \cdots + r_n)$ when the girls' preferences are cyclic? For $n = 3, 4, 5$ the values are respectively $306/3!^3$, $884224/4!^4$, $104035560000/5!^5$. When $n = 4$, the worst seven preference matrices for the girls are

$$\begin{matrix}1&2&3&4\\2&3&4&1\\3&4&1&2\\4&1&2&3\end{matrix}\quad \begin{matrix}1&2&3&4\\2&3&1&4\\3&4&1&2\\4&1&2&3\end{matrix}\quad \begin{matrix}1&2&3&4\\2&3&1&4\\3&4&2&1\\4&1&2&3\end{matrix}\quad \begin{matrix}1&2&3&4\\2&3&4&1\\3&1&4&2\\4&1&2&3\end{matrix}\quad \begin{matrix}1&2&3&4\\2&3&1&4\\3&4&1&2\\4&1&3&2\end{matrix}\quad \begin{matrix}1&2&3&4\\2&3&1&4\\3&4&1&2\\4&2&1&3\end{matrix}\quad \begin{matrix}1&2&3&4\\1&2&3&4\\3&4&1&2\\4&2&3&1\end{matrix}$$

with respective total rank sums

$$884224,\quad 879488,\quad 875264,\quad 875072,\quad 874752,\quad 874624,\quad 872192.$$

All preference matrices not isomorphic to these seven, under renumbering of boys and girls, have smaller total rank sum over all $4!^4$ preference matrices for the boys.

ACKNOWLEDGMENT

I want to thank Boris Pittel for introducing me to this problem and for patiently correcting my original misunderstanding of the definitions.

REFERENCES

1. A. M. Frieze and B. G. Pittel, ‘‘Probabilistic Analysis of an Algorithm in the Theory of Markets in Indivisible Goods,’’ *Annals of Applied Probability*, to appear.
2. D. Gale and L. S. Shapley, College admissions and the stability of marriage, *Amer. Math. Monthly* **69** (1962), 9−15.
3. R. L. Graham, D. E. Knuth, and O. Patashnik, ‘‘Concrete Mathematics,’’ Addison−Wesley, Reading, MA, 1989.
4. D. Gusfield and R. W. Irving, ‘‘The Stable Marriage Problem,’’ MIT Press, Cambridge, MA, 1989.
5. D. E. Knuth, ‘‘Sorting and Searching,’’ Addison−Wesley, Reading, MA, 1973.
6. D. E. Knuth, ‘‘Mariages Stables,’’ Les Presses de l’Université de Montréal, Montréal, 1976.
7. D. G. McVitie and L. B. Wilson, The stable marriage problem, *Commun. ACM* **14** (1971), 486−492.
8. L. S. Shapley and H. Scarf, On cores and indivisibility, *J. Math. Econ.* **1** (1974), 23−38.