Task Assignment in HFC

Introduction

Now that HFC participants have been assigned to teams, we propose to study how to assign tasks to teams. The motivation is to avoid cognitive overload for teams, while assigning tasks to teams so that 1) teams are given tasks they are skilled in, and 2) the assignment is balanced across teams? Balance is helpful for fairness, and in supporting this primary goal of avoiding cognitive overload.

In the model of team assignment:

- There are six skill categories, and every participant has a score on each skill category.
- Participants are assigned to teams. In every skill category, a team has a score computed by summing the members' scores.
- There are 12 "specialized" teams. For each of the six skills, there are two teams that have a high score for the skill.
- There are 10 "diverse" teams. They are constructed to be well-rounded across tasks.

In the model of task assignment:

- 10 tasks are released each day, over a period of 180 days.
- Each task requires the use of one skill, and the 1800 total tasks are uniformly distributed across the six distinct skills.
- Each task is to be assigned to 50% of teams.
- Tasks expire after 30 days.

To measure the effectiveness of our task assignment, we will use the metrics **average, daily total skill usage** and **average, daily balance gap**. We also define the **total task assignment**, which is the set of active tasks currently assigned to a team.

Total skill usage: If a team has a score 1.5 on skill four (based on the sum of member scores) and is assigned a task that uses skill four, then the skill usage for the task is 1.5. The **total skill usage** on a particular day is the sum of skill usage over all teams and all assigned tasks. The average, daily total skill usage is averaged across the 180 days.

Balance Gap: We do not require that all teams receive exactly the same number of tasks. The **balance gap** on a particular day is the difference between the maximum total task assignment and the minimum total task assignment across teams. The average, daily balance gap is averaged across the 180 days.

Our task assignment assumes that teams participate in all tasks assigned to them. However, the decision between task assignment methods should consider empirical participation rates when assigning additional tasks to teams. See "suggestions" and "summary" sections for further discussion.

Assigning Tasks to Teams

We explore the trade-off between **total skill usage** and **balance gap** in our task assignment algorithm. The task assignment algorithm has two steps:

1) Compute a maximally balanced assignment of new tasks, which is the assignment that minimizes the balance gap given the new tasks and the current assignment.
2) Maximize expertise subject to balance: here, we can vary the amount by which the optimal balance gap is relaxed, maximizing skill usage subject to this constraint.

We refer to the resulting assignment as the "expert assignment." Granting the expert assignment more freedom in balance gap should increase total skill usage**.** We explore what happens when we directly increase balance gap in the expert assignment, as well as explore the effect of batching multiple days' assignments together.

Details on the integer programming formulations for the "balanced" and "expert assignment" problems can be found in the January 2018 Milestone report. The solve time of these IPs is small--- under a minute even for assigning all tasks over the complete assignment period (180 days).

Increasing the Balance gap in the Expert Assignment

Suppose the optimally balanced solution results in a total task assignment with between *MIN* and *MAX* tasks per team. Then in the expert assignment, we can enforce a total task assignment between *MIN-a* and *MAX+b* per team, for a particular choice of parameters *a* and *b*. Varying both *a+b* (the total slack in balance gap) and *a, b* individually changes the tradeoff between the average balance gap/day and the total skill use/day.

For every pair of positive integers *a,b* satisfying $1 < a+b < 7$, we ran 20 trials of daily task assignment, and found the expected tradeoff between balance gap and total skill usage (Fig 1). Higher *a+b* (total slack) results in higher balance gap and higher total skill usage. If *a+b* is fixed, then the simulation results also show that having *a, b* with large $|a-b|$ results in higher balance gap and higher total skill usage.

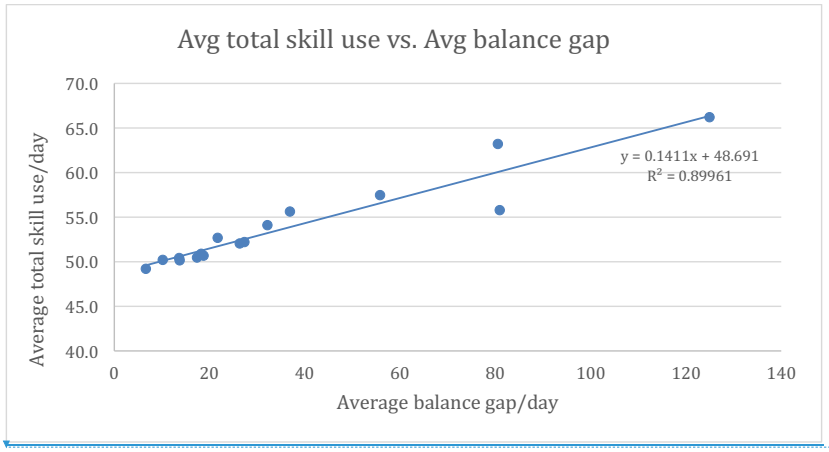


Figure 1: Relationship between average balance gap/day and average total skill usage/day

The expert assignment has two desirable properties:

1. We can estimate the effect of increasing the balance gap, allowing for a tradeoff between balance and total skill usage.

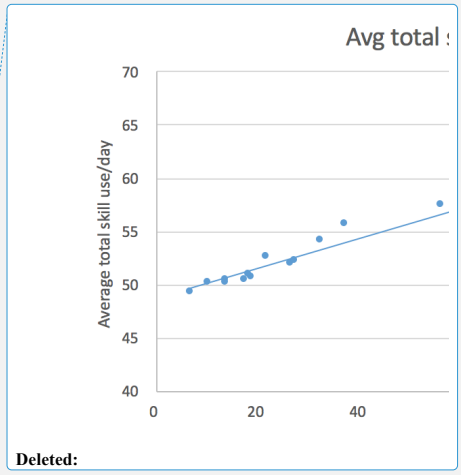
The average team receives 900 tasks over the 180 days, so if we allow the average balance gap/day to be 5% of the total task assignment/team (balance gap/day = 18), we get a 1.58 improvement in total skill usage/day. This average balance gap and improvement can be achieved with any choice of $a+b = 4$.

2. From the figure, we see that we have fine control over the balance gap: that is, a small average balance gap/day is achievable.

For a simple baseline, we can also consider a greedy algorithm that first assigns tasks to teams with skill usage > 3, and then randomly assigns the rest of the tasks. The result (balance gap/day = 386.5, total skill usage/day = 103.13) lies close to the regression line in Figure 1, which passes through (balance gap = 386.5, total skill usage = 103.15). However, this is up-right in the figure and a balance gap of 386.5 lacks any semblance of balance. Although the uniform distribution of tasks over skills might cause us to think that random assignment "evens out" balance gap over time, the greedy algorithm generates an assignment that is not well balanced.

Batching the Task Assignment

Instead of assigning tasks daily, we can batch N days' worth of tasks together, and assign them all at once, every N days. The extreme case, N = 180 days, is assignment with complete hindsight: it describes how we could have assigned the 1800 tasks so that 1) we have essentially perfect balance with regard to this total assignment, and 2) the total skill usage was maximized subject to this balance requirement.



Deleted:

Enforcing a task assignment between *MIN-1* and *MAX* for the expert solution (and thus essentially perfect balance), we obtain Figure 2. For large values of N this is not very interesting because we would need to wait too long before assigning tasks, and because the notion of balance ignores imbalance as tasks expire at different dates in between assignment rounds. But for smaller values of N, for example 2 <= N <= 5, this provides a way to improve total skill use that is complementary to the use of the balance relaxation provided by choices of a, b > 1.

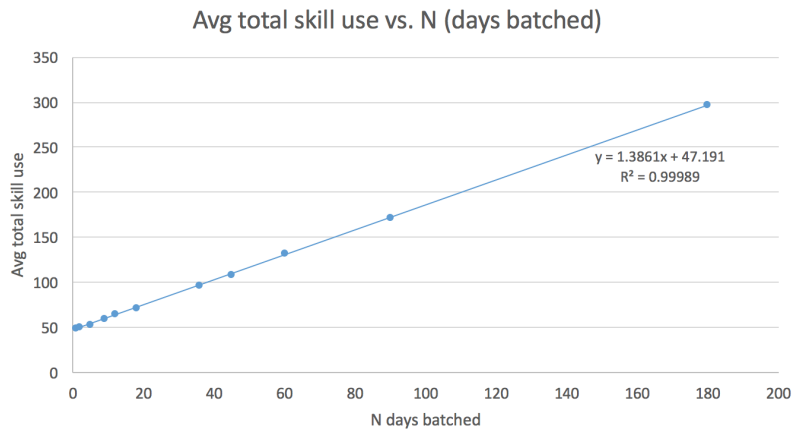


Figure 2: Relationship between days batched and average total skill usage/day

Comparing the linear regression on batching (slope = 1.38) vs. the effect of increasing the balance gap by one unit (slope = 0.14), we see that batching an additional day of assignments is approximately 10 times as effective as increasing the day-to-day balance gap by one. In particular, increasing the total slack ($a+b=2$, $a+b=3$) while batching resulted in negligible improvement in average total skill use.

Suggestions

Although the total skill usage scales linearly with an increased balance gap, and linearly with batching, the decision between increasing gap, batching, or neither should be motivated by empirical participation data, so we can properly discount skill usage when the way in which tasks are assigned to a team, and the number of tasks assigned to a team, is varied. Roughly:

- If participation in tasks is dependent on a team's skill in the task, then we should increase the balance gap, which lets us give stronger teams more tasks.

- If a delay of a few days in assignment (e.g., 2-4 days) is tolerable from a systems perspective, and assigning a lot of tasks at once does not cause other problems, then we should batch the

assignment. For example, batching into periods of 2 days results in a 2-3% increase in total skill usage.

- If participation in a task for which a team is only marginally qualified is low, then we can update the Jan2018 integer programming formulation to also consider the merits of assigning some tasks non uniformly so that not every task needs to go to 50% of the teams. We can also incorporate "high participation" and "low participation" types, to discount total skill usage differently across different teams.

Summary

Increasing the Balance Gap in the Expert Solution

- Increasing the average balance gap/day by 1 task increases the average total skill usage/day by 0.14
- If using balance gap, we recommend an average balance gap/day of 18 (5% of average total assignment), which provides an absolute improvement of 1.58 in the average total skill usage/day. This is implemented by adding 4 tasks' worth of slack to the balanced solution.

Batching the Task Assignment

- Batching one additional day of tasks increases the average total skill usage/day by an absolute amount of 1.38
- If using batching, we recommend batching 2 days. There may be concerns with delaying the assignment from batching too many days together.
- Batching and increasing the balance gap simultaneously does not perform much better than batching itself (this is because batching is almost entirely balanced and optimal).

Suggestions

- The decision between increasing the balance gap and the batching size should be determined by participation rate, or the likelihood of a team completing an additional task assigned to it. If teams always complete all tasks assigned to them, then participation rates are not a concern.
- Batching requires that teams complete additional tasks assigned to them, which is likely for low N because the tasks only expire after 30 days. If participation rates are sufficiently similar when assigning each team 10 vs. 20 tasks, then we should batch.
- Increasing balance gap lets us give more tasks to stronger teams. In the case that teams or users participate more if they are skilled in a task, then we should increase balanced gap.
- Further investigation is needed to understand how batching and increasing balance gap affect current team/task participation rates. An easy first step for the ongoing investigation is to see if participation in a task is correlated with a team or user's skill usage for the task.

Formatted: Don't add space between paragraphs of the same style, Outline numbered + Level: 1 + Numbering Style: Bullet + Aligned at: 0.25" + Indent at: 0.5"

Deleted: (but this is because the effect of batching is large relative to balance gap).

Deleted: participation rate in a marginal task.

Deleted: Are participation rates sufficiently similar if we assign 10 vs. 20 tasks? Then we should batch

Deleted: , which is balanced across teams but requires high participation rate in the marginal task.

Deleted: Are participation rates different between assigning 10 vs. 20 tasks, but positively correlated with skill level? (Are teams more likely to complete tasks they are skilled at?) Then we should increase gap, which lets us give stronger teams more tasks.

Deleted: <#>Are teams much more unlikely to complete 20 tasks than 10 tasks, with participation independent of skill level? Then we should adjust the benefits predicted by the increased gap and batching models.