# Task Assignment in HFC

## Summary

**Increasing gap in expert solution**

* Increasing average gap/day by 1 task increases average skill use/day by 0.14
* Expert solution and greedy heuristic result in the same tradeoff between total skill use and gap
* Expert solution gives us fine control over gap/day but greedy heuristic does not
* If using gap, recommend an average gap/day of 18 (5% of average total assignment), which provides 1.58 improvement in average skill use/day. Implemented by adding 4 tasks’ worth of slack to balanced solution

**Batching task assignment**

* Batching one additional day of tasks increases average skill use/day by 1.38
* If using batching, recommend batching 2 days together. Participation concerns with batching too many days together
* Batching and increasing gap simultaneously does not perform much better than batching itself

**Suggestions**

* Decision between increasing gap and batching should be determined by the participation rate in a marginal task.
* Are participation rates sufficiently similar if we assign 10 vs. 20 tasks? Then we should batch, which is balanced across teams but requires high participation rate in marginal task.
* 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.
* 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.

## 

## Introduction

Now that HFC participants have been assigned to teams, we propose a question in task (IFP) assignment. How can we assign tasks to teams so that 1) teams are given tasks they are skilled in and 2) the assignment is balanced across teams?

In our model of team assignment:

* There are 6 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 6 skills, there are 2 teams that have a high score in the skill.
* There are 10 diverse teams. They are constructed to be well-rounded across tasks.

In our 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 6 skills.
* Each task is assigned to 50% of teams.
* Tasks expire after 30 days.

To measure the effectiveness of our task assignment, we will use the metrics total **skill usage** and **gap**. We will also define **total task assignment.**

**Total skill usage**: If a team has score 1.5 on skill four and is assigned a task that uses skill four, then their skill usage for the task is 1.5. The **total skill usage** is the sum of skill usage over all teams and all assigned tasks.

**Total task assignment:** The set of active tasks assigned to a team. It is the tasks that are either 1) newly assigned or 2) previously assigned but unexpired.

**Gap:** We do not require that all teams receive exactly the same number of tasks. The **gap** is the difference between the maximum total task assignment and the minimum total task assignment across teams.

## Matching tasks to teams

We explore the trade-off between **total skill usage** and **gap** in our task assignment, which has two steps:

1. Compute balanced solution, which is the assignment that minimizes gap.
2. Assign expert solution, which allows for slightly more gap than the balanced solution and then maximizes total skill usage.

Granting the expert assignment more freedom in gap should increase total skill usage**.** We explore what happens when we directly increase gap in the expert solution and enforce balance less frequently by batching multiple days’ assignments together.

Details on the balanced and expert IP can be found in the Jan2018 Milestone. Runtime is under a minute for every simulation of the total assignment period (180 days).

## Increasing gap in the expert solution

Suppose the balanced solution results in total task assignment between *MIN* and *MAX* per team. Then in the expert assignment, we can enforce a total task assignment between *MIN-a* and *MAX+b* per team for given *a* and *b*. Varying both *a+b* (total slack in gap) and *a,b* individually affects average gap/day and 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 a positive linear relationship between gap and total skill use (Fig 1). Higher *a+b* (total slack) results in higher gap and skill. If *a+b* is fixed, then having *a,b* with large *|a-b|* results in higher gap and skill.

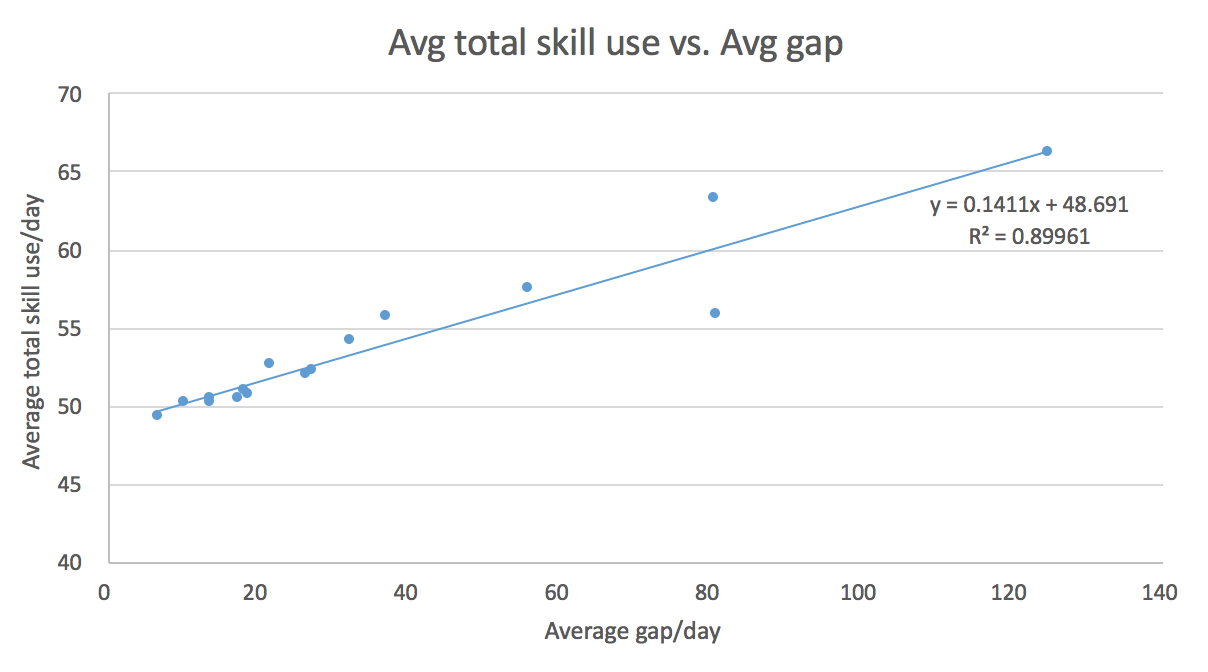


Figure 1: Positive linear relationship between average gap/day and average total skill use/day

Our expert solution has two desirable properties:

1. We can compute the benefit of increasing gap.

The average team receives 900 tasks over the 180 days, so if we allow the average gap/day to be 5% of the total task assignment/team (gap/day = 18) rather than the optimally balanced gap/day (gap/day = 6.7), we get a 1.58 improvement in total skill use/day. This average gap and improvement can be achieved with any slack satisfying *a+b = 4*.

1. We have fine control over gap: that is, a small average gap/day is implementable.

Consider a simple greedy algorithm that first assigns tasks to teams with skill usage > 3 and then randomly assigns the rest of the tasks. The result (gap = 386.5, total skill usage = 103.13) lies surprisingly close to the regression line in Figure 1, which passes through (gap = 386.5, total skill usage = 103.15). However, a 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” gap over time, the large gap created by the greedy algorithm demonstrates the difficulty of maintaining low gap.

## Batching task assignment

Instead of assigning tasks daily, we can batch N days’ worth of tasks (10N tasks) together and assign them all at once. We will only require balance every N days, rather than daily. The extreme case, N = 180 days, is assignment with complete hindsight and balance: it describes how, in hindsight, we could have assigned the 1800 tasks so that 1) gap after 180 days was balanced and 2) total skill usage was maximized.

Enforcing a task assignment between *MIN-1* and *MAX* for the expert solution, we obtained Figure 2. Gap decreases as N increases, and for N=1 we have gap = 6.66. However, average gap/day calculations are artificially low because the assignment are made only every N days. For example, when N = 180, the gap is 0 the first 179 days (because no tasks are assigned) and only positive on the 180th day, when balance is enforced.

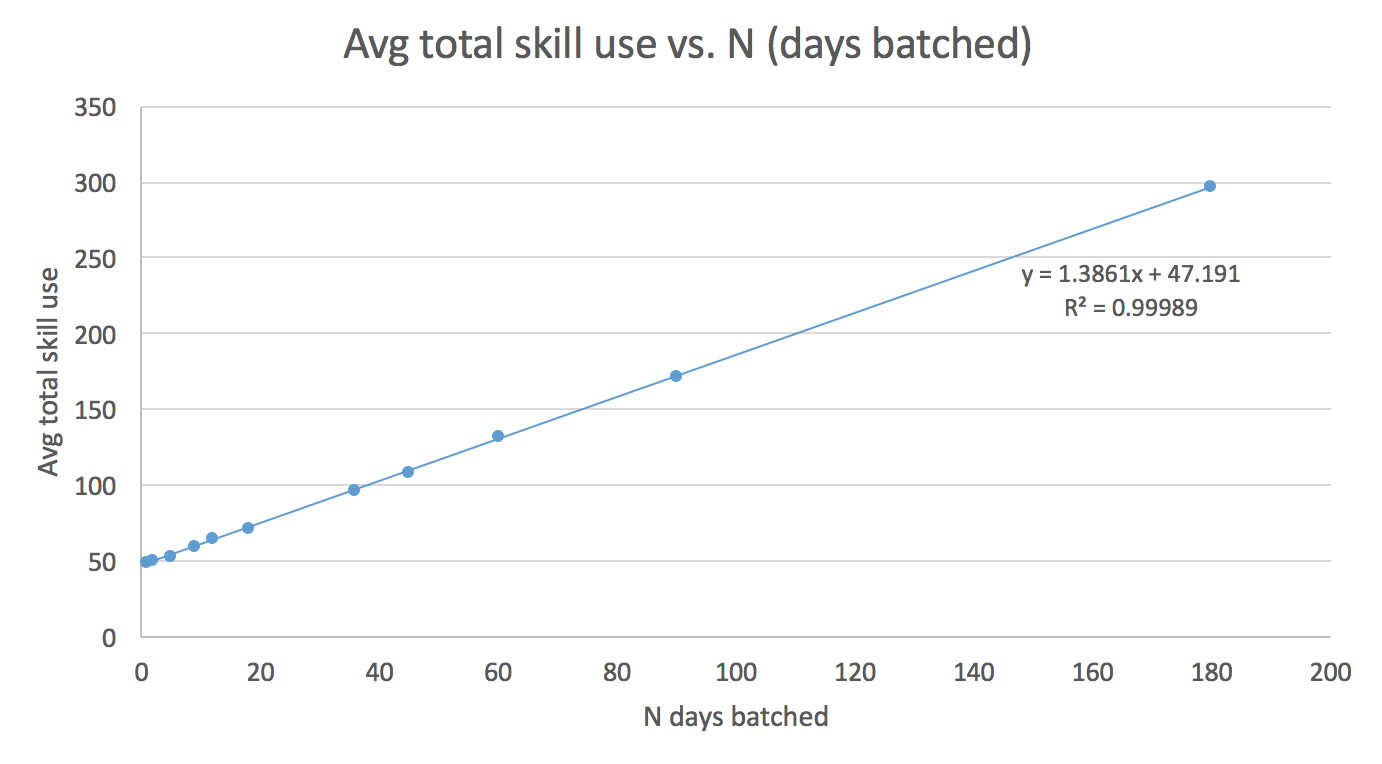


Figure 2: Positive linear relationship between days batched and and average total skill use

Comparing the linear regression on batching (slope = 1.38) vs. day-to-day gap (slope = 0.14), we see that batching an extra day of assignments is approximately 10 times as effective as increasing the day-to-day gap by 1. 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 use theoretically scales linearly with days batched, we might expect that assigning more tasks to a team causes a lower rate of participation. The decision between increasing gap, batching, or neither should be motivated by empirical participation data, so we can properly discount skill usage when more tasks are assigned to a team. Roughly:

* If participation in tasks is dependent on a team’s skill in the task and marginal participation in an extra task is reasonably high, then we should increase gap, which lets us give stronger teams more tasks.
* If participation in 10 assigned tasks occurs at the same rate as participation in 20 assigned tasks, then we should batch. Note that batching 2 days results in a 2-3% increase in total skill usage, so the decrease in participation from 10 tasks to 20 must be negligible.
* If participation in a marginal task is low, then we can update the Jan2018 IP to discount total skill usage based on the total task assignment to the team. This can be added as a linear constraint. We can also incorporate “high participation” and “low participation” types to discount total skill usage different across teams.