When is crowdsourcing a good way to do work?

**Single-sourcing**
- Players submit “bids” to enter a crowdsourcing contest
- Players: Workforce is global and decentralized
- Tasks: Project-centric rather than job-centric work
- How can one find the best person for the job and incentivize him to attain peak performance?

**Multi-sourcing**
- With three or more players, only top two players are active

---

**To Crowdsource or not to Crowdsource?**

Gireeja Ranade* and Lav R. Varshney*

*Department of EECS, University of California, Berkeley
*IBM T. J. Watson Research Labs

---

**Single Task Model**
Crowdsourcing works well to complete a single task when:
- Managers are uncertain about the skills of workers
- Workers are diverse in their task skills
- Workers have low default effort levels

BUT, with many similar hardworking workers, crowdsourcing is not worth it.

---

**Multiple Task Model**

- 2n players, 2n tasks – Multi-player, multi-item auction
- 2 different types of tasks and players
- nt players, n tasks of each type

**When is crowdsourcing a good way to do work?**

“There is this misconception that you can sprinkle crowd wisdom on something and things will turn out for the best. That’s not true. It’s not magic.”

---

**All-Pay Auction Model**
Players submit “bids” to enter a crowdsourcing contest

\[ A = \text{prize value} \]

\[ c_i = \text{cost of effort for player } i, \; i = 1, 2. \text{ WLG, player 1 is stronger, i.e. } c_2 > c_1 \]

\[ x_i = \text{bid submitted by player } i, \; i = 1, 2 \]

**Player utilities given by**

\[ EU_1 = A P(x_1 > x_2) - x_1c_1 \]

\[ EU_2 = A P(x_2 > x_1) - x_2c_2 \]

Equilibrium strategy (Hillman and Riley, 1989)

- Stronger player bids \( U(0, A/c_2) \)
- Weaker player bids \( U(0, A/c_2) \) with probability \( c_1/c_2 \) and bids 0 otherwise
- With three or more players, only top two players are active

---

**Task Taxonomy**

Model utility gained from task-completion

**Conditions for positive task utility**

**Task-designer utility:** function of task, players, bid & reward

\[ U_{\text{task}} = f(\epsilon) - A \]

- **Selective** (e.g. software component design)
  \[ f(\epsilon) = \max(x_1, x_2, \ldots, x_n) \]

- **Integrative** (e.g. information aggregation, idea generation) perhaps with coordination overhead
  \[ f(\epsilon) = a \sum_{i=1}^{n} x_i, \quad \text{or} \quad f(\epsilon) = a \sum_{i=1}^{n} x_i - y_n, \; a, y > 0 \]

- **Involve market creation** (e.g. X PRIZE)

Utility depends on externalities, not player effort (e.g. no. of players, \( \phi \))

\[ f(\epsilon) = \alpha n + \beta \]

- For instance, for a basic selective two-player task:
  \[ EU_{\text{task}} = \frac{d_2x_1x_2}{c_2} - A \]

**Single Task Model**

Capture information difference between players & manager

- **Distance measure (cost)** between tasks & players, \( d_1 < d_2 \)
- **Manager** knowledge of players’ skills is noisy
- **Players** have better self knowledge
- **Managerial assignment** requires external motivation
- \( \theta \) denotes base effort
- **Competitions provide endogenous motivation**

\[ EU_{\text{opt}} = A \left( \frac{\theta}{d_1} - 1 \right) \]

\[ EU_{\text{man}} = A P(\text{right}) \frac{\theta}{d_1} + P(\text{wrong}) \frac{\theta}{d_2} - 1 \]

\[ EU_{\text{ca}} = \frac{d_2x_1c_2}{d_2} - A \]

**Multiple Task Model**

- **2n players, 2n tasks** – Multi-player, multi-item auction
- **2 different types of tasks and players**
- nt players, n tasks of each type

**With multiple, diverse tasks to complete:**

- Crowdsourcing contests can perform as well as optimal assignment of workers to tasks for non-specialized tasks (i.e. if enough skilled players are available).
- **BUT**
  - Crowdsourcing contests can provide very low utility even with many strong players and just one weak player.
  - Crowdsourcing contests can perform badly for highly specialized tasks. Instead of pulling out highly-skilled workers from a crowd, crowdsourcing could lead to mediocre performance by everyone.