Designing a Human Computing Algorithm

Jérôme Waldispühl, McGill University
Computational problems

- Multiplication: two numbers $\rightarrow$ product
- Sorting: set of objects $\rightarrow$ set of objects sorted
- Medical diagnosis: x-ray, lab tests $\rightarrow$ diagnosis
- Object recognition: image $\rightarrow$ tag
- Translation: source sentence $\rightarrow$ target sentence
- Editing: text $\rightarrow$ corrected text
- Planning: goal, constraints $\rightarrow$ sequence of actions
What is an algorithm?

“A finite set of rules which gives a sequence of operations for solving a specific type of problem” such that:

- **Input**: One or more inputs
- **Output**: One or more outputs which have a specific relation to the input(s)
- **Finiteness**: It must terminate after a finite number of steps.
- **Effectiveness**: Each operation needs to be basic
- **Definiteness**: Each step must be well defined and unambiguous.

(Knuth, 1973)
How to structure human-computation programs?
Operations & Controls

Sequence:

Operation 1

Condition 1

Operation 1

Operation N

Repetition:

Condition 1

Operation 1

Operation 1

Operation N

Selection:

Condition 1

Operation 1

Operation 2

Parallel:

Operation 1

Operation N

(Kowalski, 1979)
Humans in the loop

1. Humans perform basic operation
2. Humans execute the control structure
3. Program synthesis and collaborative programming
Human as operators: Quicksort

function `quicksort`(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function `pivot`(A)
    return randomIndex(A);

function `compare`(x, pivot)
    return (x < pivot)
Human as operators: Subjective sorting with Quicksort

```plaintext
function quicksort(A)
    initialize empty lists L and G
    if (length(A) ≤ 1)
        return A
    pivot = A.remove(find_pivot(A));
    for x in A
        if compare(x, pivot)
            L.add(x)
        else
            G.add(x)
    return concatenate(quicksort(L), pivot, quicksort(G))

function pivot(A)
    return randomIndex(A);

function compare(x, pivot)
    return human_compare(x, pivot)
```

Amazon Mechanical Turk: Game with a purpose
Human as operators: Graph Coloring

1 conflict in your immediate neighborhood.
A thick line indicates a conflict that must be resolved.
A thin line is shown when color choices do not conflict.
Overall progress toward a solution:

1 conflict in your immediate neighborhood.
A thick line indicates a conflict that must be resolved.
A thin line is shown when color choices do not conflict.
Overall progress toward a solution:

1 conflict in your immediate neighborhood.
A thick line indicates a conflict that must be resolved.
A thin line is shown when color choices do not conflict.
Overall progress toward a solution:

(Kearns et al., 2006)
Human as controllers: Simulated Annealing

\[ s \leftarrow s_0; e \leftarrow E(s) \]
\[ sbest \leftarrow s; ebest \leftarrow e \]
\[ k \leftarrow 0 \]
\[ \text{while Human\_Decision}(e,k) \text{ do} \]
\[ T \leftarrow \text{temperature}(k/k_{\text{max}}) \]
\[ s_{\text{new}} \leftarrow \text{neighbour}(s) \]
\[ e_{\text{new}} \leftarrow E(s_{\text{new}}) \]
\[ \text{if } P(e, e_{\text{new}}, T) > \text{random()} \text{ then} \]
\[ s \leftarrow s_{\text{new}}; e \leftarrow e_{\text{new}} \]
\[ \text{if } e_{\text{new}} < e_{\text{best}} \text{ then} \]
\[ sbest \leftarrow s_{\text{new}}; ebest \leftarrow e_{\text{new}} \]
\[ k \leftarrow k + 1 \]
\[ \text{return } sbest \]

HIT or GWAP
Program synthesis

• Crowdforge
• Turk~o~matic

Both use the Amazon Mechanical Turk
Amazon Mechanical Turk

A framework to crowdsource human-computation tasks.

<table>
<thead>
<tr>
<th>HIT: Human Intelligent Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find Legal Categories for Law Firms</strong></td>
</tr>
<tr>
<td><strong>Requester:</strong> Jason Lavallee</td>
</tr>
<tr>
<td><strong>Time Allotted:</strong> 60 minutes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Inv. B.2</strong></th>
<th><strong>View a HIT in this group</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requester:</strong> rohjit0d</td>
<td><strong>HIT Expiration Date:</strong> Feb 10, 2013 (3 weeks 6 days)</td>
</tr>
<tr>
<td><strong>Time Allotted:</strong> 48 minutes</td>
<td><strong>HITs Available:</strong> 20216</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Search: Keywords on Google.com (US)</strong></th>
<th><strong>View a HIT in this group</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requester:</strong> CrowdSource</td>
<td><strong>HIT Expiration Date:</strong> Jan 13, 2014 (52 weeks)</td>
</tr>
<tr>
<td><strong>Time Allotted:</strong> 16 minutes</td>
<td><strong>HITs Available:</strong> 14973</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Write: Creative Article Title</strong></th>
<th><strong>View a HIT in this group</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requester:</strong> CrowdSource</td>
<td><strong>HIT Expiration Date:</strong> Jan 13, 2014 (52 weeks)</td>
</tr>
<tr>
<td><strong>Time Allotted:</strong> 15 minutes</td>
<td><strong>HITs Available:</strong> 10549</td>
</tr>
</tbody>
</table>
Amazon Mechanical Turk

Find the meaning of the following hashtag

- For this hashtag below, enter the its definition or its meaning. The definition should explain what that hashtag means.
- Include links if it refers to a website entering the full address, e.g. "Dubai is an emirate in the United Arab Emirates #UAE. A city within this emirate is called Dubai also." is a correct definition in case of #Dubai hashtag.
- If you do not know the definition just return the HIT so somebody else can enter the correct definition. If you fail to enter the right definition the HIT will be rejected and it will affect your stats.
- One Hashtag may have different meanings. For example #IMU stands for "I Miss You" and also for "Inbound Marketing University". Please enter as many meanings as you can.

Hashtag: #KFUPM

Enter your hashtag definition here:

You must ACCEPT the HIT before you can submit the results.
Program synthesis: Crowdforge

- **Partition tasks**: Split a problem into sub-problems (one to many)
- **Map tasks**: Solve a small unit of work (one to one)
- **Reduce tasks**: Combine multiple results into one (many to one)

http://smus.com/crowdforge/
Writing papers with Crowdforge

Partition (outline)

Map (facts)
- The Empire State Building celebrated its 75th Anniversary on May 1, 2006.
- ...most popular ones include the Empire State Building, the Statue of Liberty and the Grand Central Terminal.
- The Statue of Liberty has become an American symbol of freedom and welcome to the immigrants.

Reduce (paragraph)

Ask most people who plan to travel to New York City what they want to see while they are there and invariably you will hear about the top tourist attractions: the Empire State Building, the Statue of Liberty, and the Grand Central Terminal, with the Empire State Building probably coming in as number one on the list of "must see" for visitors to the city. No wonder: the Empire State Building has a long history, having celebrated its seventy-fifth anniversary on May 1, 2006. Yet the Statue of Liberty is also a popular tourist destination.

http://smus.com/crowdforge/
Current limits of Crowdforge

• Currently it does not support iteration or recursion,
• It assumes that complex works can be broken up into relatively small and independent pieces.

(Kittur et al., 2011)
Program synthesis: Turk-0-matic

• Similar to Crowdforge but include a recursive mechanism,
• Additional workflow visualization and editing capabilities.

System:
1. A meta-workflow uses the crowd to assist in the design of the workflow, and to execute it,
2. An editable workflow visualization enables requesters to observe and manage work being executed by the crowd.

(Kulkarni et al., 2012)
Turk~o~matic workflow

Please create a new
split

Choose a theme for a
solve Theme-Mechanical Tur

Choose a free bloggi
solve Blogger (formerly BI

Open a new blog about
split

Open a new blog about
solve mechanical turk tran

Choose a name for th
decide

Write a short openin
decide

Write a post about M
decide

Change the color sch
decide
Workflow (1): Sub-division

Break down the task written in red.

Instructions: We are dividing a large task among several workers on Mechanical Turk. This is an experiment to see how complicated tasks can be shared between multiple workers on Mechanical Turk. Your job is to help us plan how this work should be divided.

Here is the task you are asked to divide:

Write a 3-paragraph essay about crowdsourcing

Do not solve this task yourself. Please break the task down into 2 or more simpler steps. Write each step in a box below. You can add more steps.

Each step you suggest will be posted to Mechanical Turk again for another Turker to do. Make sure each step will make sense to another Turker.

Here is what makes a good answer:

- Every step is a complete sentence or set of instructions.
- Each step contains all information required to do the task.
- Every step explains clearly what a Turker should do.
- Each step can be understood by itself without reading the original task written in red.

Tips:

- You can ask Turkers to host images and pictures on other sites, like http://imgur.com or http://youtube.com.

Your work will be checked for correctness before being approved.

Step 1

Step 2

Add Step | Remove Step
Vote on the work of other Turkers

We gave several Turkers the following task and asked them to break it down into a set of smaller tasks:

Write a 3-paragraph essay about crowdsourcing

They gave the following breakdowns:

Turker 1:

Step 1: Visit online databases and libraries to find academic articles about crowdsourcing.
Step 2: Read the articles and consider what three points you can develop about crowd surfing.
Step 3: Read the articles and highlight the data you can use.

Turker 3:

Step 1: Write a 3-paragraph essay
Step 2: Write about crowdsourcing

Choose the best breakdown for the task from among the ones given. Specifically, use this criteria:

- Is every step a complete sentence or set of instructions?
- Does every step explain clearly what a Turk should do?
- Can you understand what each step is asking you to do without reading the original task in blue?

Answer carefully: your work will only be approved if your answer matches the majority of other Turkers.
Workflow (3): Solution

Solve a simple task

Instructions: We are dividing a large task among several workers on Mechanical Turk. This is an experiment to see how to break down large tasks. You are asked to do a small part of a large task that was planned by other workers.

The overall task: Write a 3-paragraph essay about crowdsourcing

Your task:

Visit online databases and libraries to find academic articles about crowdsourcing.

Your instructions: Please do this task and enter the solution in the box at the bottom of this page. You are free to include links to other images or videos you have uploaded online. If the instructions do not make sense, please take a look at the overall plan below and take your best guess.

Optional: click here to email us feedback about this HIT.

Here is the plan made by other workers:

- The overall task: Write a 3-paragraph essay about crowdsourcing
  - Step 1. Visit online databases and libraries to find academic articles about crowdsourcing. (this is your step)
  - Step 2. Read the articles and consider what three points you can develop about crowd surfing.
  - Step 3. Read the articles and highlight the data you can use.
Workflow (4): Merging

Your goal is to find a solution to the following task highlighted in orange by combining the answers of other Turkers:

Write a 3-paragraph essay about crowdsourcing

Other Turkers have suggested that this task can be broken into the steps written in green below. These steps have already been solved by other Turkers. Their solutions are written below.

Please combine the solutions written below into a single solution to the task written in orange. You should modify the solutions as necessary to better solve the task written in orange.

Sub-task 1: Visit online databases and libraries to find academic articles about crowdsourcing.
Solution to sub-task 1: Crowdsourcing has endless possibilities especially in the hands of creators, inventors and the curious public. So far in the past, Makerbot has invited members of the 3D printing community to...
Editing Workflows (1)

Task Graph for Job 63

- Write a 3-paragraph
  - split
- Collect more information
  - decide
- Think about the topic
  - split
- Think about the topic
  - solve
  - The word may be a noun
- Please write your own
  - decide
- Write it down.
  - split

Edit Task Details

Task: Think about the topic that you want to write.
Status: split

- Edit Task
- Edit Solution
- Add Subtask
- Delete Node
Editing Workflows (2)

Instructions:
List three main topics about crowdsourcing.

Solution Strategy:
○ Let Workers Decide
○ Solve Task
○ Split Task

Solution: Will be recomputed based on your edit.
Editing Workflows (3)

Task Graph for Job 63

See Tree | Back to list

- Write a 3-paragraph
- split
- List three main topics
- solve
- Collect more information
- decide
- Please write your own
- decide
How to implement algorithms using human-computation?
Programming framework: Turkit

http://groups.csail.mit.edu/uid/turkit/
Basic functions

- **once(func, name)**: Calls the function only once and store the results.
- **fork(func, name)**: Executes func in a pseudo-thread.
- **join()**: Makes sure all the forks have finished.

REST requests to Mechanical Turk using once:
- **mturk.prompt(message, params)**
- **mturk.vote(message, options, votesNeeded, params)**
- **mturk.survey(message, options, count, params)**
Example (1) : Generating ideas

```javascript
ideas = []
for (var i = 0; i < 5; i++) {
    idea = mturk.prompt(
        "What’s fun to see in New York City?
        Ideas so far: " + ideas.join(", "))
    ideas.push(idea)
}

ideas.sort(function (a, b) {
    v = mturk.vote("Which is better?", [a, b])
    return v == a ? -1 : 1
})
```
Example (2) : Sorting with Turkit

```plaintext
quicksort(A)
    if A.length > 0 then
        pivot ← A.remove(once A.randomIndex())
        left ← new array
        right ← new array
        for x in A do
            if compare(x, pivot) then
                left.add(x)
            else
                right.add(x)
        quicksort(left)
        quicksort(right)
        A.set(left + pivot + right)

compare(a, b)
    hitId ← once createHIT(...a...b...)
    result ← once getHITResult(hitId)
    return (result says a < b)
```
How to evaluate human-computation algorithms?
Effectiveness: Time complexity

Critical in many aspects of human-computation algorithms (E.g. text translation, image description).

3 types of complexity:
• Run-time
• Clock time
• Discovery time
Effectiveness: Query complexity

Number of HIT required to execute the algorithm.

2 decisions:
• Who should execute the task? (i.e. Human or computer)
  o Humans do all tasks,
  o Sample tasks to humans to build a training set for A.I.
  o Sample *Selected* tasks to humans (i.e. Active learning)
• How to distribute the tasks? (i.e. single or multiple workers)
Effectiveness: Cost

- Cost to complete the program with paid platforms,
- Mostly motivated by budget constraints,
- Does not concern GWAP!
References

Human Computation
Edith Law, Luis von Ahn
Morgan & Claypool Publishers