The Simplified Vancouver Algorithm

Notation:

- Graders indexed by i.
- Submissions indexed by j.
- Given grader i, S_i denotes the set of submissions graded by i.
- Given submission j, T_j denotes the set of graders who graded j.

Data: Grades g_{ij}, of grader i for submission j.

Variables:

- For each submission j:
  - Grade estimate $\hat{g}_j$.
  - Variance estimate $\hat{v}_j$.
- For each grader i: grader variance estimate $\hat{u}_i$.

Initialization: For all i, $\hat{u}_i^0 = 1$. [Or whatever you want, really.]

Algorithm: Iterate the following until convergence:

1. Update submission variance estimates:
$$\hat{v}_j^{(t+1)} = \left(\sum_{i \in T_j} \frac{1}{\hat{u}_i^{(t)}} \right)^{-1}$$
2. Update grade estimates:
$$\hat{g}_j^{(t+1)} = \hat{v}_j^{(t+1)} \sum_{i \in T_j} \frac{1}{\hat{u}_i^{(t)}} g_{ij}$$
3. Update user variance estimates:
$$\hat{u}_i^{(t+1)} = \left(\sum_{j \in S_i} \frac{1}{\hat{v}_j^{(t+1)}} \right)^{-1} \sum_{j \in S_i} \frac{1}{\hat{v}_j^{(t+1)}} \left(g_{ij} - \hat{g}_j^{(t+1)} \right)^2$$

Note: The above pseudocode is superficially very different from the version presented in the CrowdGrader paper for two reasons:

1. The version in the paper updates estimates of (for example) each submission j's grade $\hat{g}_j^{(t+1)}$ by dropping out j's variables in the computation of $\hat{u}_i^{(t)}$ for each $i \in T_j$ (similarly for the $\hat{v}_j$s and $\hat{u}_i$s). We do not believe this makes a significant difference.
2. Because the above version avoids the dropout procedure from the paper, it does not need to use message passing.