Algorithm 1: Serial Double Greedy Alg.

$A \leftarrow \emptyset, B \leftarrow V$
for $i \in \{1, \ldots n\}$ **do**
  $\Delta_A \leftarrow \max(F(A \cup i) - F(A), 0)$
  $\Delta_B \leftarrow \max(F(B \backslash i) - F(B), 0)$
  Draw $u_i \sim Unif(0, 1)$
  if $u_i < \frac{\Delta_A}{\Delta_A + \Delta_B}$ **then** $A \leftarrow A \cup i$
  else $B \leftarrow B \backslash i$

Algorithm 2: Parallel Double Greedy Alg.

$A \leftarrow \emptyset, B \leftarrow V$
for $p \in \{1, \ldots, P\}$ **do in parallel**
  $i \leftarrow p$
  while $i < n$ **do**
    $(\Gamma_i, \delta_i) \leftarrow T_i(A, B)$
    $(A, B) \leftarrow$ Validate(δ_i, Γ_i)
    $i \leftarrow$ AtomicInc(i)

Algorithm 3: Validate the Transaction

Input: The predicate Γ on the state A, B
Input: The operation δ_i on element $i \in V$
if $\Gamma(A, B)$ *is false* **then**
  $\Delta_A = F(A \cup i) - F(A)$
  $\Delta_B = F(B \backslash i) - F(B)$
  if $u_i < \frac{\Delta_A}{\Delta_A + \Delta_B}$ **then** $\delta_i \leftarrow +1$
  else $\delta_i \leftarrow -1$
if $\delta_i = +1$ **then** $A \leftarrow A \cup i$
else $B \leftarrow B \backslash i$

Algorithm 4: Coordinate Free Transaction T_i

$\Delta_A \leftarrow \max(F(A \cup i) - F(A), 0)$
$\Delta_B \leftarrow \max(F(B \backslash i) - F(B), 0)$
Draw $u_i \sim Unif(0, 1)$
if $u_i < \frac{\Delta_A}{\Delta_A + \Delta_B}$ **then** (True, +1)
else (True, -1)

Algorithm 5: Concurrency Control T_i

$\Delta_A \leftarrow \max(F(A \cup i) - F(A), 0)$
$\Delta_B \leftarrow \max(F(B \backslash i) - F(B), 0)$
Draw $u_i \sim Unif(0, 1)$
$j \leftarrow i + 1$
if $u_i < \frac{\Delta_A}{\Delta_A + \Delta_B}$ **then**
  `// Construct the tight bound`
  $\bar{A} := A \cup \{i + 1, \ldots, j\}$
  $\underline{B} := B \backslash \{i + 1, \ldots, j\}$
  while $u_i < \frac{\Delta_{\bar{A}}}{\Delta_{\bar{A}} + \Delta_{\underline{B}}}$ **do**
    $j \leftarrow j + 1$
  $(\Gamma(A', B') = A' \subset \bar{A}$ & $\underline{B} \subset B'$, +1)
else
  [JG: finish other condition] (True, -1)

4 CF2G Bidirectional Greedy Algorithm

[XP: Provide more intuition for what the CF2G algorithm is doing.]

Algorithm 6 is the CF2G parallel bidirectional greedy algorithm for unconstrained submodular maximization.[1] The CF2G algorithm closely resembles the serial algorithm, but the elements $e \in V$ are no longer processed in a fixed order. Thus, the sets A, B are replaced by "bounds" $\hat{A}, \hat{B}$, where $\hat{A}$ is a subset of the "true" A and $\hat{B}$ is a superset of the "actual" B on each iteration. These bounding sets allow us to compute bounds $\Delta_+^{\max}, \Delta_-^{\max}$ which approximate Δ_+, Δ_- from the serial algorithm. We now formalize this idea.

We order the elements $e \in V$ according to the time at which Algorithm 6 line 7 is executed. Let $\iota(e)$ be the position of e in this total ordering on elements. This ordering allows us to define monotonically non-decreasing sets $A^i = \{e' : e' \in A, \iota(e') < i\}$, and monotonically non-increasing sets $B^i = A^i \cup \{e' : \iota(e') \geq i\}$. These "true" sets A^i, B^i provide a serialization against which we can compare the CF2G algorithm; in this serialization, Algorithm 1 computes:

$$\Delta_+(e) = F(A^{\iota(e)-1} \cup i) - F(A^{\iota(e)-1}), \qquad \Delta_-(e) = F(B^{\iota(e)-1} \backslash e) - F(B^{\iota(e)-1}).$$

Note that in Algorithm 6, lines 5 and 6 may be executed in parallel with lines 9 and 10. Hence, $\Delta_+^{\max}(e)$ and $\Delta_-^{\max}(e)$ (lines 5 and 6) may be computed with conflicting versions of $\hat{A}$ and $\hat{B}$. Denoting these versions by $\hat{A}_e$ and $\hat{B}_e$, Algorithm 6 computes:

$$\Delta_+^{\max}(e) = F(\hat{A}_e \cup e) - F(\hat{A}_e), \qquad \Delta_-^{\max}(e) = F(\hat{B}_e \backslash e) - F(\hat{B}_e).$$

[1] We present only the parallelized probabilistic versions of [1]. Both parallel algorithms can be easily extended to the deterministic version of [1]; the CF2G algorithm can also be extended to the multilinear version of [1].