Review of the Submission

Summary

This paper presents an algorithm for the k-means problem in the Massively Parallel Computation (MPC) model. The algorithm computes a constant-factor approximation. The stated round complexity is $O(\log\log n \cdot \log\log\log n)$. It operates in the fully scalable MPC setting, utilizing $O(n^\sigma)$ local memory per machine and $O(n^{1+\epsilon})$ global memory for arbitrarily small constants $\sigma, \epsilon > 0$. This provides a constant-factor approximation for the general k-means problem in $o(\log n)$ rounds in the MPC model.

The approach is based on the Jain and Vazirani (JV01) framework, reducing k-means to Facility Location (FL) via a Lagrangian Multiplier Preserving (LMP) approximation. The core technical component is a parallel algorithm for LMP FL adapted to the MPC model.

The methodology involves using Locality-Sensitive Hashing (LSH) to create a sparse graph representation of the metric space. The algorithm then executes a parallelized primal-dual approach for FL, which involves directly estimating dual variables rather than iterative doubling, and handling inconsistencies by identifying "problematic clients." A significant component is the computation of a ruling set on a dependency graph of facilities. Due to the complexity of computing $(O(1), O(1))$-ruling sets in the low-memory MPC model, the algorithm uses a hybrid approach: an adapted Luby's algorithm covers a large fraction of the weight within $O(1)$ distance, and the algorithm of [KPP20] covers the remainder within $O(\log\log\log n)$ distance.

Errors and Improvements

Major Issues

1. **Invalid Constant Relationships in Lemma 4.2 Proof (Pages 16-17):** The proof of Lemma 4.2 relies on specific relationships between constants defined in Eq. (14), assuming $\Gamma \geq 5$.

   - In the first part of Case 1 (Page 17), the derivation concludes $\geq \lambda$ based on the assumption $C_A \geq Q \cdot C_D^+$. The constants are defined as $C_A = 8\Gamma^8$, $Q = 8\Gamma^4$, and $C_D^+ = 8\Gamma^4$. Thus, $Q \cdot C_D^+ = 64\Gamma^8$. The required inequality $8\Gamma^8 \geq 64\Gamma^8$ does not hold, invalidating this step.
   - In the latter part of Case 1 (Page 17), the final inequality requires the derived expression to be $\leq \eta \cdot \alpha_{c,0}$. The coefficient derived is $12C_D^+ + 12\gamma_2^2 C_D^+ + 3\eta\frac{C_D^+}{C_D^- Q}$. Substituting the defined constants ($\eta = 8000\Gamma^{12}$, etc.) yields $96\Gamma^4 + 7776\Gamma^{12} + 12000\Gamma^{10}$. This must be $\leq 8000\Gamma^{12}$, which simplifies to $96\Gamma^4 + 12000\Gamma^{10} \leq 224\Gamma^{12}$. This inequality does not hold for $\Gamma \geq 5$ (e.g., if $\Gamma = 5$, $12000(5^{10}) > 224(5^{12})$).

2. **Incorrect Definition/Application of Relaxed Triangle Inequality (Eq. 2):** The relaxed triangle inequality is defined in Eq. (2) (Page 5) as: $\sum_{i=1}^{\ell} \text{cost}(x_{i-1}, x_i) \leq \ell \cdot \text{cost}(x_0, x_\ell)$ (assuming x_s is a typo for x_ℓ). The paper states this holds for squared Euclidean distances (Page 8). This is incorrect. (E.g., 1D points $x_0 = 0, x_1 = 10, x_2 = 0$; LHS=200, RHS=0).

   The standard property for squared Euclidean distances is the reverse: $\text{cost}(x_0, x_\ell) \leq \ell \cdot \sum_{i=1}^{\ell} \text{cost}(x_{i-1}, x_i)$. The proofs rely on this standard property. For example, in Lemma 4.2 (Page 17), Eq. (2) is cited to justify $\text{cost}(c, f) \leq 3 \cdot (\text{cost}(c, c') + \text{cost}(c', c'') + \text{cost}(c'', f))$. The definition in Eq. (2) needs correction.

   Furthermore, Eq. (2) is defined only for sequences alternating between clients and facilities. The application on Page 17 uses the sequence (c, c', c'', f), which does not alternate.

Minor Issues

1. **Runtime Inconsistency in Lemma 5.9 (Pages 35, 37):** The statement of Lemma 5.9 claims a runtime of $O(t \cdot \log\log n)$ rounds. The analysis within the proof (Page 37) concludes

that the overall time for computing the set S is $O(t \log\log\log n + \log\log n \cdot \log\log\log n)$. The lemma statement should be consistent with the derivation in the proof.

2. **Missing Factor in Lemma 5.2 Edge Bound (Page 29):** Lemma 5.2 states an edge bound of $|E_D| \leq 5\ln(n)/p_1$. The algorithm uses $t = 5\ln(n)/p_1$ hash functions, and the proof (Page 31) notes that each adds at most $n-1$ edges. The bound should be $O(n \cdot \ln(n)/p_1)$. The factor of n is missing in the lemma statement.

3. **Incorrect Probability Expression in Lemma 5.2 Proof (Page 31):** The proof states: "The probability that x and y are not connected by a path of length ≤ 2 is therefore at most $1-(1-p_1)^t \geq 1-e^{-p_1 \cdot t} = \cdots = 1/n^5$." This expression is mathematically incorrect. The probability of not being connected is $(1-p_1)^t$, which is upper bounded by $e^{-p_1 t} = 1/n^5$.