Suggested MS&E310 Project 2: Simplex on GPU

$$\min \quad c^T x = \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \quad a_1 x = \sum_{j=1}^{n} a_{1j} x_j = b_1$$

$$a_2 x = \sum_{j=1}^{n} a_{2j} x_j = b_2$$

$$\ldots$$

$$a_m x = \sum_{j=1}^{n} a_{mj} x_j = b_m$$

$$x \geq 0$$

→

$$\min \quad c^T x$$

$$\text{s.t.} \quad Ax = b,$$

$$x \geq 0$$

(Primal) Simplex Method

1. Start with a Basic Feasible Solution with basis B and compute basic variables $\mathbf{x}_B = (A_B)^{-1}\, \boldsymbol{b} (\geq \mathbf{0})$, and let non-basic variables $\mathbf{x}_N = \mathbf{0}$.

2. Compute **shadow price** vector: $\mathbf{y}^T = \mathbf{c}^T_B (A_B)^{-1}$

3. Calculate the **reduced cost** vector for non-basic variables

$$r_N = c^T_N - y^T A_N$$

If the reduced cost for every non-basic variable is nonnegative, then STOP: OPTIMAL

4. Dantzig Rule: select the most negative reduced cost variable, say x_e as the entering variable with column $\boldsymbol{A}_e$, compute $(A_B)^{-1} \boldsymbol{A}_e$ and use the minimum ratio to decide the outgoing variable (row). If the min-ratio is infinity, then STOP: declare UNBOUNDED

5. Update new basis (B) matrix inverse $(A_B)^{-1}$; or perform the pivot operations to update the tableau.

Go to Step 1

Simplex Method on GPUs

1. Compute $\mathbf{x}_B = (A_B)^{-1}\, \boldsymbol{b}(\geq \mathbf{0})$ in $O(m^2)$ operations, and it can be done in $O(m)$ operations and $O(m)$ communications if rows of $(A_B)^{-1}$ are distributed on m GPUs

2. compute **shadow price** vector $\mathbf{y}^T = \mathbf{c}^T{}_B (A_B)^{-1}$ in $O(m^2)$ operations, and can be done in $O(m)$ operations and $O(m)$ communications if columns of $(A_B)^{-1}$ are distributed on m GPUs .

3. Calculate the **reduced cost** vector for non-basic variables $\boldsymbol{r}_N = \boldsymbol{c}^T{}_N - \boldsymbol{y}^T A_N$ in $O(m(n-m))$ operations, and can be done in $O(m)$ operations and $O(m)$ communications if columns of A_N are distributed on $(n-m)$ GPUs similarly.
On CPU: If the reduced cost for every non-basic variable is nonnegative, then STOP: OPTIMAL

4. On CPU: select the most negative reduced cost variable, say x_e as the entering variable with column $\boldsymbol{A}_e$, compute vector $(A_B)^{-1} \boldsymbol{A}_e$ in $O(m)$ operations and $O(m)$ communications on GPU again. Then on CPU: using the minimum ratio to decide the outgoing variable (row). If the min-ratio is infinity, then STOP: UNBOUNDED

5.Update new basis (B) and matrix inverse $(A_B)^{-1}$ in $O(m^2)$ operations, and can be done in $O(m)$ operations and $O(m)$ communications if rows or columns of $(A_B)^{-1}$ are distributed on m GPUs.