Now we are ready to construct supply rates for stochastic finite-sum methods. In many situations, the analysis of stochastic finite-sum methods only require simple supply rates that can be obtained by manipulating the quadratic constraints covered in the last two lectures. First we will focus on SAGA-like methods. Then we will briefly discuss SVRG which is another important finite-sum method.

### 9.1 Supply Rates for SAGA-Like Methods

Recall that SAGA-like methods can be represented as $F_u(G, \Delta)$ where $G$ is a jump system and the operator $\Delta$ maps $v$ to $w$ as

$$w_k = \begin{bmatrix} \nabla f_1(v_k) \\ \nabla f_2(v_k) \\ \vdots \\ \nabla f_n(v_k) \end{bmatrix}$$

(9.1)

For this operator $\Delta$, we want to construct pointwise quadratic constraints on the input/output pair $(v, w)$:

$$\begin{bmatrix} v_k - v^* \\ w_k - w^* \end{bmatrix}^T M \begin{bmatrix} v_k - v^* \\ w_k - w^* \end{bmatrix} \leq 0,$$

(9.2)

where $M$ is a symmetric matrix, and $(w^*, v^*)$ are determined by the fixed points of the feedback interconnection $F_u(G, \Delta)$. For SAGA, we know $v^* = x^*$ and $w^* = [\nabla f_1(x^*)^T \cdots \nabla f_n(x^*)^T]$ where $\nabla f(x^*) = \frac{1}{n} \sum_{k=1}^n \nabla f_i(x^*) = 0$.

Again, if we know $v_k - v^* = C(\xi_k - \xi^*)$, the above quadratic constraint (9.2) just gives the following supply rate condition

$$\begin{bmatrix} \xi_k - \xi^* \\ w_k - w^* \end{bmatrix}^T \left( \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T M \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \right) \begin{bmatrix} \xi_k - \xi^* \\ w_k - w^* \end{bmatrix} \leq 0.$$

Hence we just focus on how to obtain the quadratic constraint (9.2). Various assumptions on $f_i$ and $f$ can be converted into inequalities in the form of (9.2). Now let’s look at a few concrete examples.
• Assumption 1: \( f_i \) is \( L \)-smooth and \( m \)-strongly convex. In this case, we know
\[
\begin{bmatrix}
  v_k - v^* \\
  \nabla f_i(v_k) - \nabla f_i(v^*)
\end{bmatrix}^T
\begin{bmatrix}
  2mLI & -(L + m)I \\
  -(L + m)I & 2I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  \nabla f_i(v_k) - \nabla f_i(v^*)
\end{bmatrix}
\leq 0. \tag{9.3}
\]

We need to make use of the following key relation:
\[
w_k - w^* =
\begin{bmatrix}
  \nabla f_1(v_k) - \nabla f_1(v^*) \\
  \nabla f_2(v_k) - \nabla f_2(v^*) \\
  \vdots \\
  \nabla f_n(v_k) - \nabla f_n(v^*)
\end{bmatrix}
\tag{9.4}
\]
which leads to \( \nabla f_i(v_k) - \nabla f_i(v^*) = (e_i^T \otimes I)(w_k - w^*) \) where \( e_i \) is a vector whose \( i \)-th entry is 1 and all other entries are 0. Therefore, we have
\[
\begin{bmatrix}
  v_k - v^* \\
  \nabla f_i(v_k) - \nabla f_i(v^*)
\end{bmatrix} =
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & e_i^T \otimes I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix} \tag{9.5}
\]
Substituting the above equation into (9.3) leads to
\[
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix}^T
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & e_i^T \otimes I
\end{bmatrix}^T
\begin{bmatrix}
  2mLI & -(L + m)I \\
  -(L + m)I & 2I
\end{bmatrix}
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & e_i^T \otimes I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix} \leq 0
\]
Therefore, we can just choose \( M = \begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & e_i^T \otimes I
\end{bmatrix} \).

• Assumption 2: \( f \) is \( L \)-smooth and \( m \)-strongly convex. In this case, we know
\[
\begin{bmatrix}
  v_k - v^* \\
  \nabla f(v_k) - \nabla f(v^*)
\end{bmatrix}^T
\begin{bmatrix}
  2mLI & -(L + m)I \\
  -(L + m)I & 2I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  \nabla f(v_k) - \nabla f(v^*)
\end{bmatrix}
\leq 0. \tag{9.6}
\]
Based on (9.4), we have \( \nabla f(v_k) - \nabla f(v^*) = \frac{1}{n}(e^T \otimes I)(w_k - w^*) \) where \( e := \sum_{i=1}^{n} e_i \) is a vector whose entries are all 1. Therefore, we have
\[
\begin{bmatrix}
  v_k - v^* \\
  \nabla f(v_k) - \nabla f(v^*)
\end{bmatrix} =
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & \frac{1}{n}e^T \otimes I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix}
\]
Substituting the above equation into (9.6) leads to
\[
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix}^T
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & \frac{1}{n}e^T \otimes I
\end{bmatrix}^T
\begin{bmatrix}
  2mLI & -(L + m)I \\
  -(L + m)I & 2I
\end{bmatrix}
\begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & \frac{1}{n}e^T \otimes I
\end{bmatrix}
\begin{bmatrix}
  v_k - v^* \\
  w_k - w^*
\end{bmatrix} \leq 0.
\]
Therefore, we can just choose \( M = \begin{bmatrix}
  I & 0_{p \times (np)} \\
  0 & \frac{1}{n}e^T \otimes I
\end{bmatrix} \).
- Assumption 3: \( f_i \) is \( L \)-smooth but may not be convex. In this case, we know
\[
\begin{bmatrix}
v_k - v^*
\end{bmatrix}^T \begin{bmatrix}
\nabla f_i(v_k) - \nabla f_i(v^*)
\end{bmatrix} \begin{bmatrix}
-L^2 I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
v_k - v^*
\end{bmatrix} \leq 0. \tag{9.7}
\]

Similarly, we can substitute (9.5) into (9.7) and get
\[
\begin{bmatrix}
v_k - v^*
\end{bmatrix}^T \begin{bmatrix}
I & 0_{p \times (np)} \\
0 & e_1^T \otimes I
\end{bmatrix} \begin{bmatrix}
-L^2 I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
v_k - v^*
\end{bmatrix} \leq 0
\]

Therefore, we can just choose \( M = \begin{bmatrix}
I & 0_{p \times (np)} \\
0 & e_1^T \otimes I
\end{bmatrix} \begin{bmatrix}
-L^2 I & 0 \\
0 & I
\end{bmatrix} \begin{bmatrix}
I & 0_{p \times (np)}
\end{bmatrix}. \]

- Assumption 4: \( f \) satisfies the “one-point convexity” condition:
\[
\begin{bmatrix}
v_k - x^*
\end{bmatrix}^T \begin{bmatrix}
2mLI & -(L + m)I \\
-(L + m)I & 2I
\end{bmatrix} \begin{bmatrix}
v_k - x^*
\end{bmatrix} \leq 0. \tag{9.8}
\]

Notice the difference between (9.8) and (9.6) is that \( v^* \) is allowed to be any point in (9.6). Due to the facts \( v^* = x^* \) and \( \frac{1}{n}(e^T \otimes I)w^* = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^*) = 0 \), we still have \( \nabla f(v_k) - \nabla f(v^*) = \frac{1}{n}(e^T \otimes I)(w_k - w^*) \). Similar to before, we can just choose
\[
M = \begin{bmatrix}
I & 0_{p \times (np)} \\
0 & \frac{1}{n}e^T \otimes I
\end{bmatrix} \begin{bmatrix}
2mLI & -(L + m)I \\
-(L + m)I & 2I
\end{bmatrix} \begin{bmatrix}
I & 0_{p \times (np)}
\end{bmatrix}.
\]

How to use the above quadratic constraints? Depending on the assumptions on \( f_i \) and \( f \), we can choose multiple \( M_j \) \((j = 1, \ldots, J)\) accordingly and formulate the following LMI
\[
\sum_{i=1}^n \left( p_i \begin{bmatrix}
A_i^T PA_i - \rho^2 P & A_i^T PB_i \\
B_i^T PA_i & B_i^T PB_i
\end{bmatrix} \right) \leq \sum_{j=1}^J \lambda_j \begin{bmatrix}
C & 0 \\
0 & I
\end{bmatrix}^T M_j \begin{bmatrix}
C & 0 \\
0 & I
\end{bmatrix},
\]

where the positive definite matrix \( P \) and non-negative scalers \( \lambda_j \) are decision variables. When the assumptions on \( f_i \) and \( f \) change, typically one only needs to modify \( M_j \) accordingly. The convergence rates of SAGA and several standard finite-sum methods (SDCA, Finito, etc) can be obtained using the above quadratic constraints and LMI formulations. However, the convergence rate proof of SAG is more subtle and requires the so-called Lure-Postnikov-type Lyapunov function. We will talk about that in the next lecture.

### 9.2 Supply Rates for SVRG

Now we briefly discuss SVRG that is built upon the idea of variance reduction. Originally we model the SG method as \( F_i(G, \Delta) \) where \( \Delta \) maps \( v \) to \( w \) as \( w_k = \nabla f_i(v_k) \). We directly developed the supply rate condition for \( \Delta \) and obtain some condition in the form of \( \mathbb{E} S \leq C \).
where $C$ is a positive constant. A physical interpretation is that the stochastic gradient $\nabla f_i(v_k)$ keeps on supplying energy into the system and hence the system is not going to converge to its fixed point. Now we take a closer look. We can actually rewrite the SG method as

$$x_{k+1} = x_k - \alpha (\nabla f_i(x_k) - \nabla f_i(x^*)) - \alpha \nabla f_i(x^*)$$

If we choose $\xi_k = x_k$, $v_k = x_k$, $w_k = \left[ \begin{array}{c} \nabla f_i(v_k) - \nabla f_i(x^*) \\ \nabla f_i(x^*) \end{array} \right]$, $A = I$, $B = \left[ \begin{array}{cc} -\alpha I & -\alpha I \end{array} \right]$, and $C = I$, we obtain a new feedback representation for the SG method. Now the input $w_k$ has two entries. Actually it is trivial to construct a supply rate condition to couple the first entry of $w_k$ with $x_k - x^*$. For example, if $f_i$ is $L$-smooth and $m$-strongly convex, the following inequality holds in an almost sure sense

$$\left[ \begin{array}{c} v_k - x^* \\ \nabla f_i(v_k) - \nabla f_i(x^*) \end{array} \right]^T \left[ \begin{array}{cc} 2mLI & -(m + L)I \\ -(m + L)I & 2I \end{array} \right] \left[ \begin{array}{c} v_k - x^* \\ \nabla f_i(v_k) - \nabla f_i(x^*) \end{array} \right] \leq 0. \quad (9.9)$$

Hence the first entry of $w_k$ is not delivering energy into the system. The troublesome term is the second entry of $w_k$. The term $\nabla f_i(x^*)$ keeps on delivering energy into the system.

SVRG modifies the second entry of $w_k$ as $\nabla f_i(x^*) - \nabla f_i(x_0) + \nabla f(x_0)$. Now this input depends on the initial state $x_0$. One will be able to obtain a supply rate condition in the form of $\mathbb{E}S \leq L\|x_0 - x^*\|^2$. SVRG is an epoch-based algorithm and at the beginning of each epoch it will update $x_0$ as the last (or average) iterate of the last epoch. Notice for each epoch, one needs to evaluate one full gradient $\nabla f(x_0)$. Hence the selection of the epoch length is going to affect the performance of SVRG. Within one epoch, $x_0$ is a fixed vector. As more epochs are run, $x_0$ gets closer to $x^*$. The supplied energy eventually decreases to 0 as $x_0$ converges to $x^*$. This is a rough physical explanation for the convergence mechanism of SVRG. The dissipation inequality approach can be applied to analyze SVRG and its accelerated variant Katyusha. We omit the details here.