Parallel ADMM for robust quadratic optimal resource allocation problems

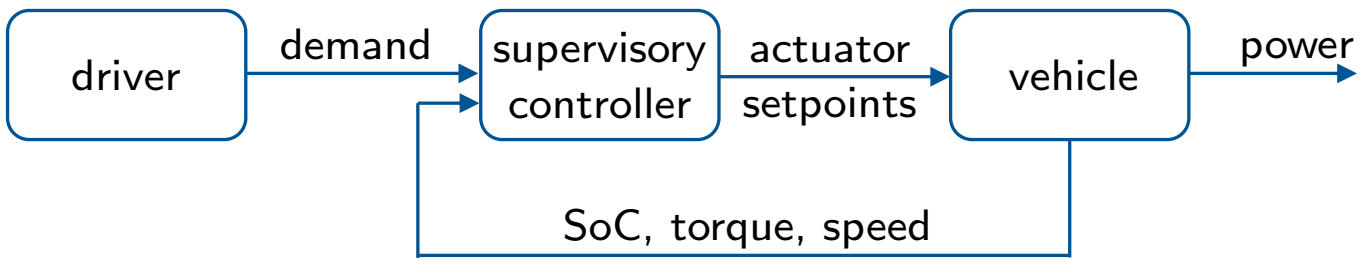
Zawar Qureshi Sebastian East Mark Cannon

University of Oxford

July 10, 2019

Motivation: PHEV energy management

Optimize power delivered by
i.c. engine and electric motor
while meeting driver demand

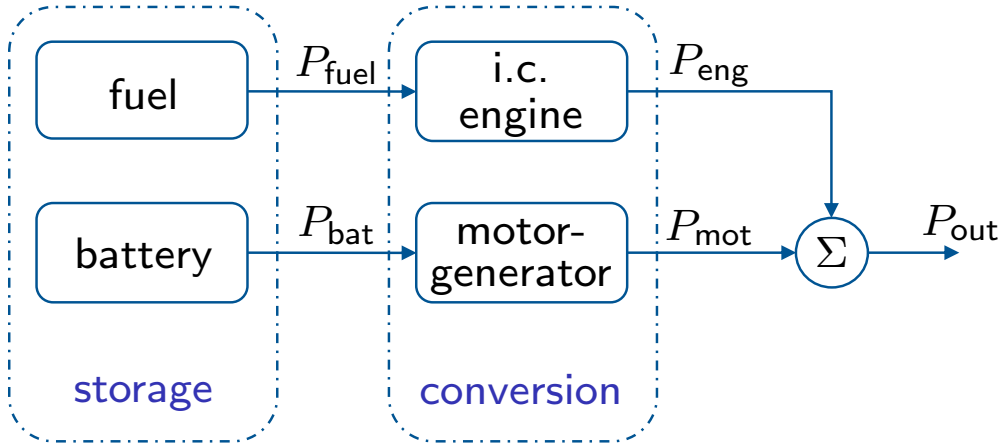


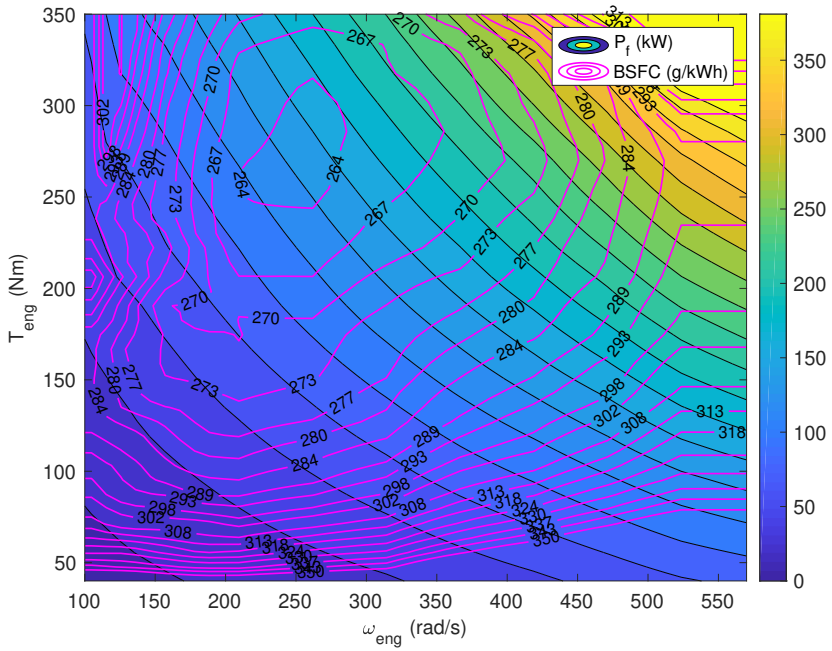
Minimize fuel consumption over a future horizon given

- ⋆ limits on energy capacity (battery SoC, fuel), power flows, torques
- ⋆ predicted demand from planned route, traffic, driver behaviour

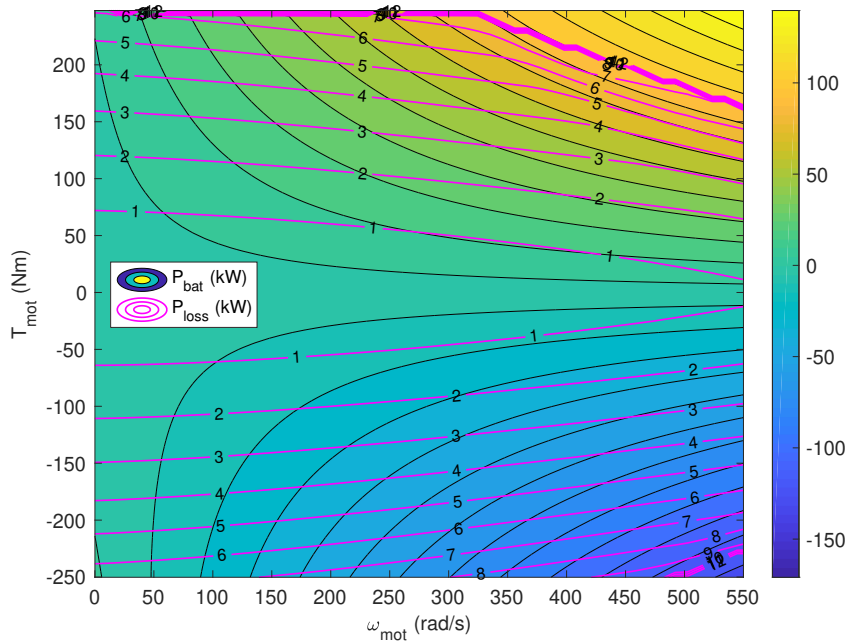
Motivation: PHEV powertrain model

Piecewise quadratic maps fitted to fuel map & electrical loss map





Engine fuel map

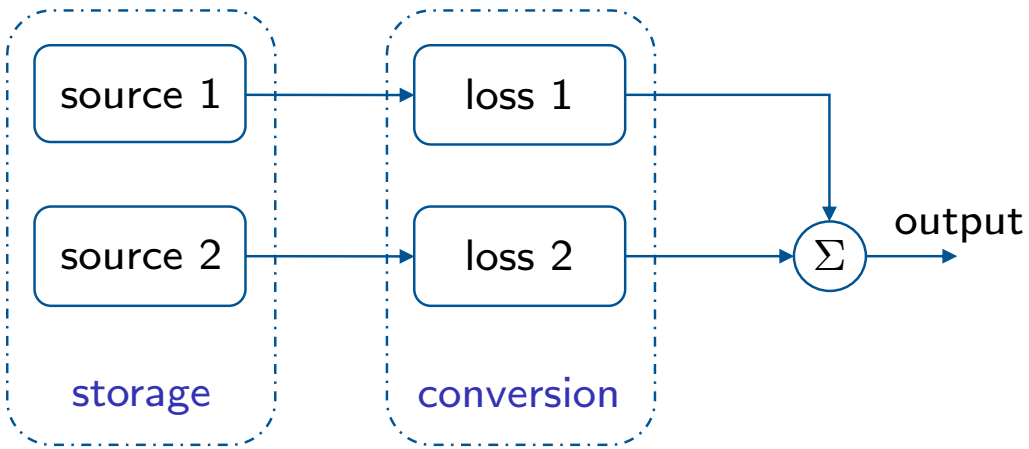


Electrical losses

Motivation: PHEV powertrain model

This paper:

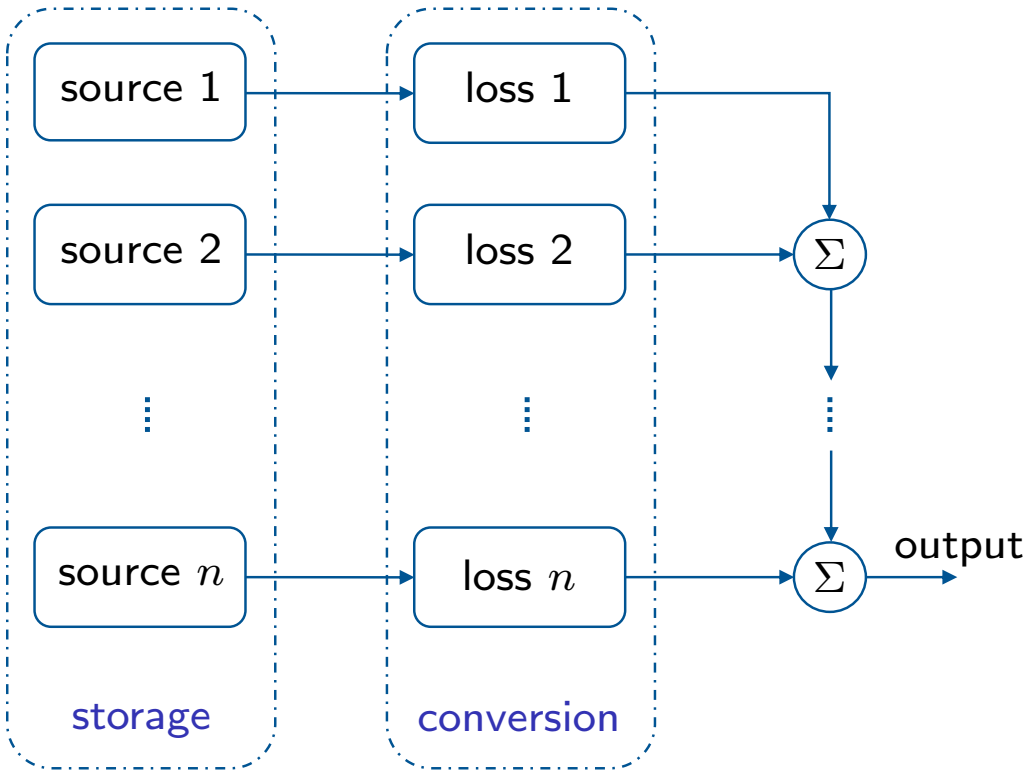
- ▷ Stochastic demand & robust optimization
- ▷ Multiple resources
- ▷ Quadratic losses & quadratic costs
- ▷ Parallel ADMM implementation



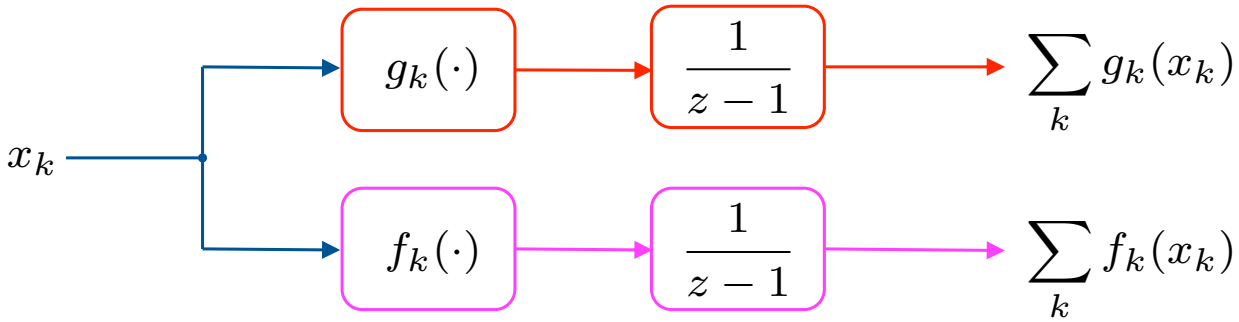
Motivation: PHEV powertrain model

This paper:

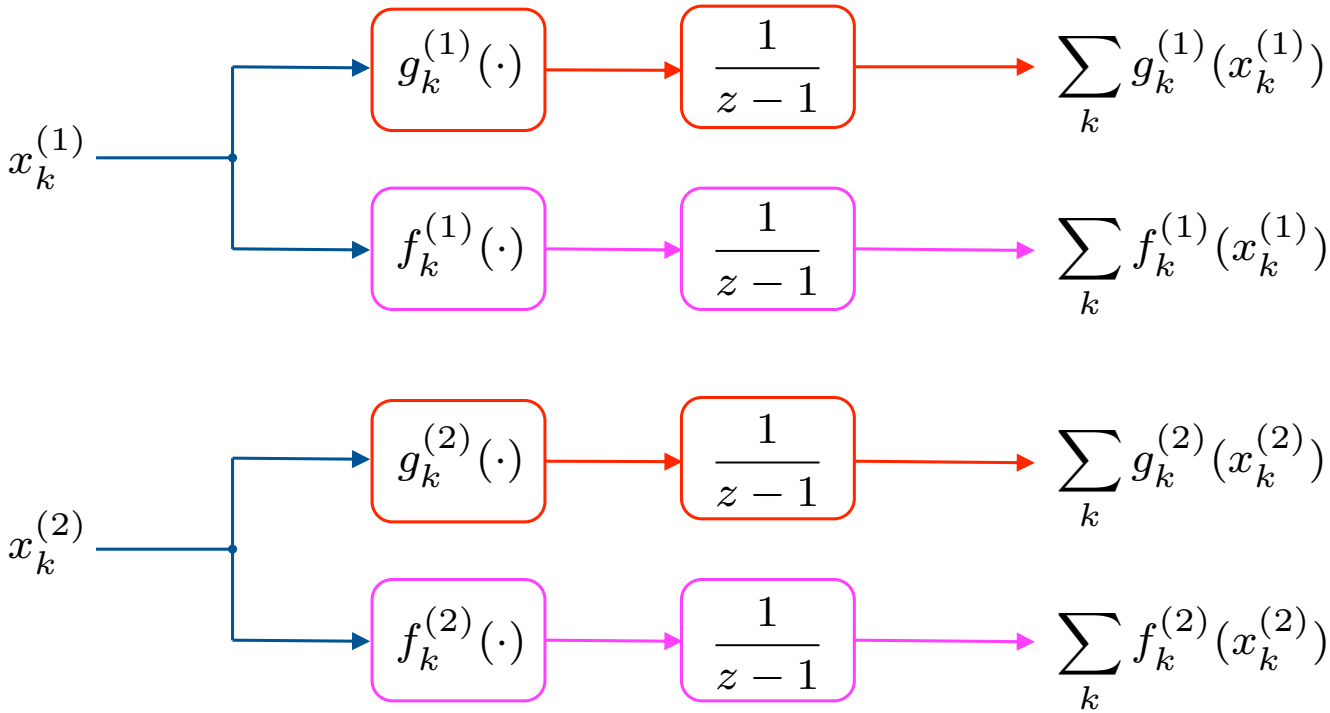
- ▷ Stochastic demand & robust optimization
- ▷ Multiple resources
- ▷ Quadratic losses & quadratic costs
- ▷ Parallel ADMM implementation



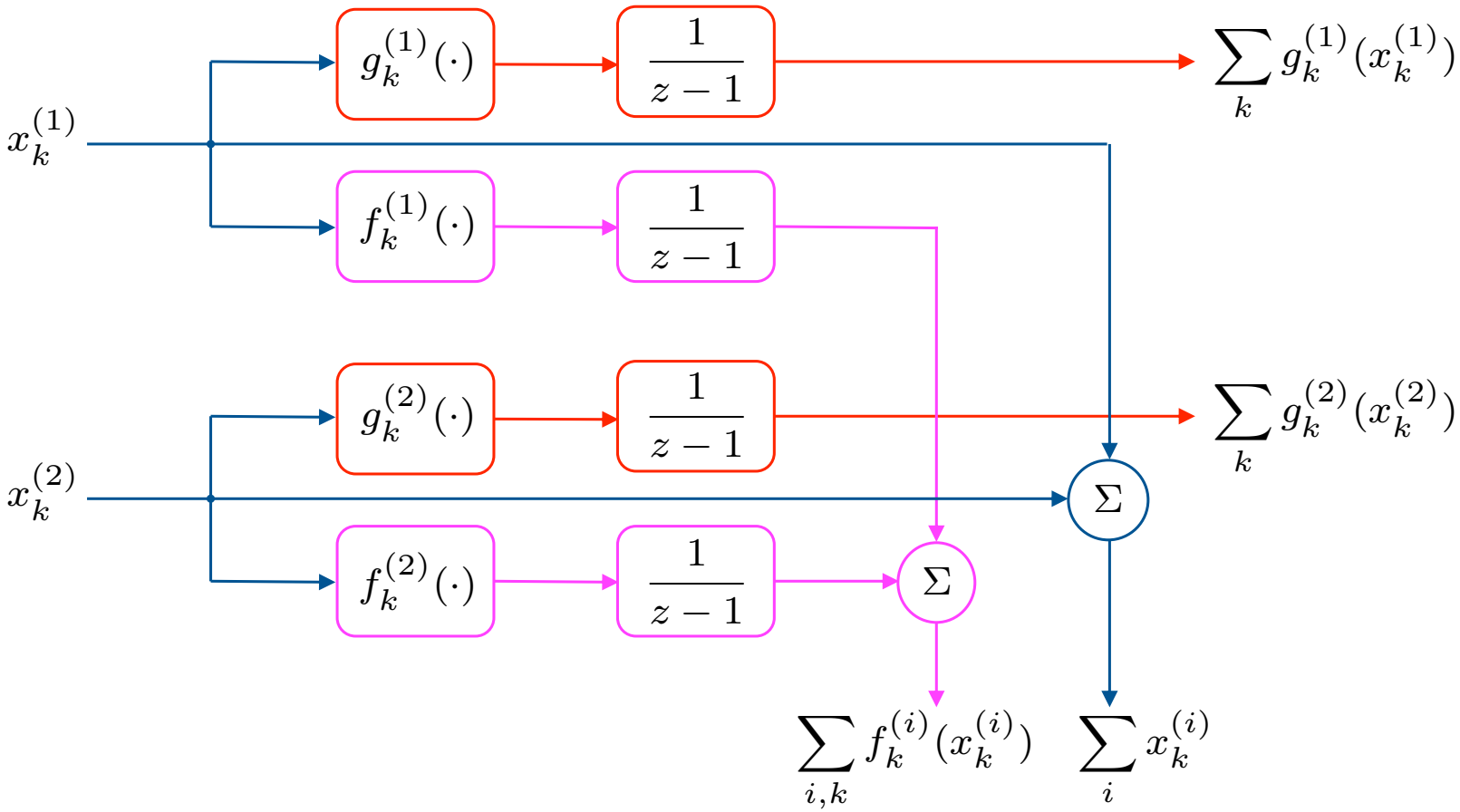
Resource allocation problem definition



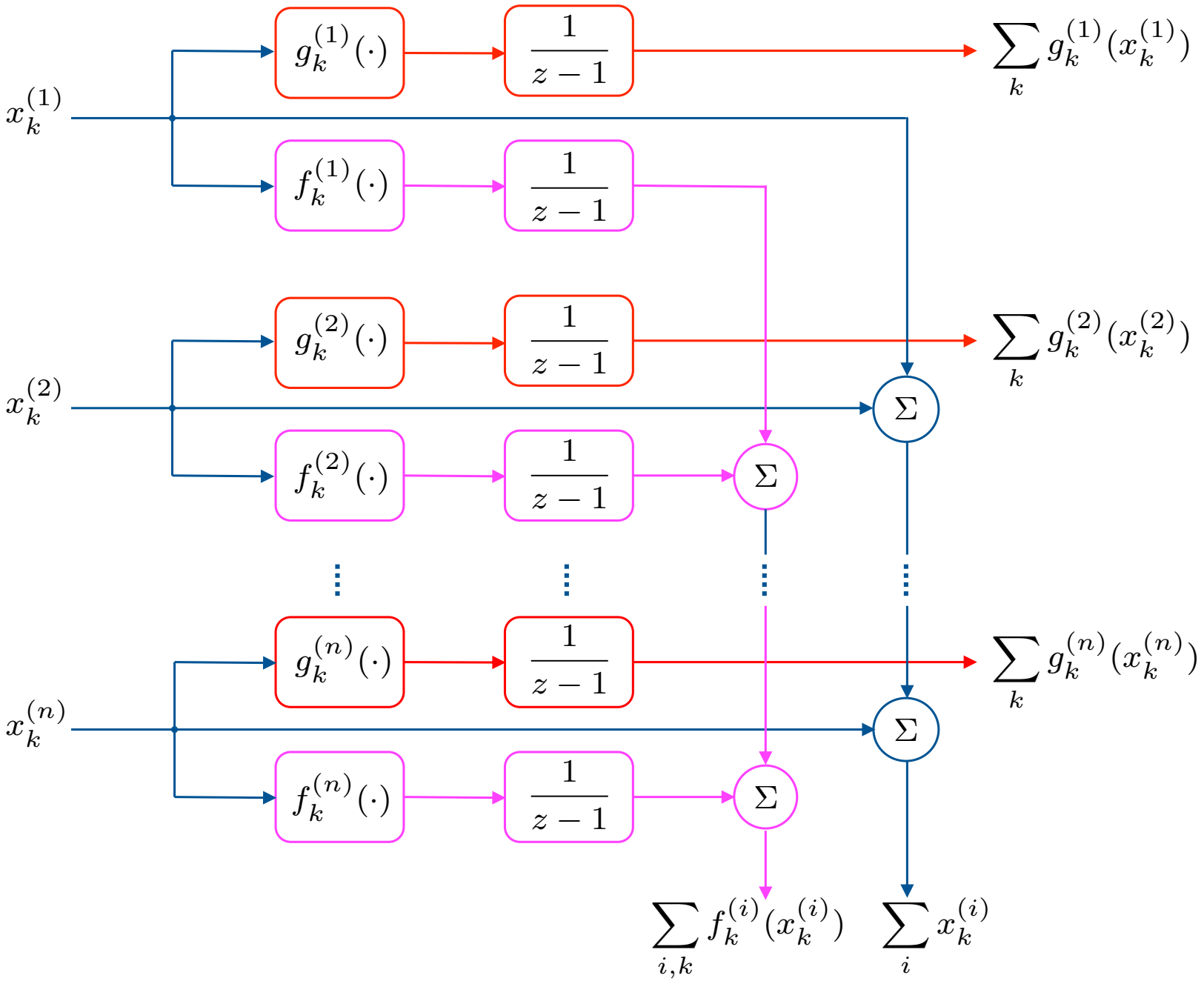
Resource allocation problem definition



Resource allocation problem definition



Resource allocation problem definition



Resource allocation problem definition

If demand sequence: $\{y_1, \ldots, y_n\}$ is known

Optimal allocation for no uncertainty

$$\begin{aligned}
&\underset{x_k^{(i)} \in [\underline{x}_k^{(i)}, \bar{x}_k^{(i)}]}{\text{minimize}} && \sum_{i,k} f_k^{(i)}(x_k^{(i)}) && && \leftarrow \text{total cost} \\
&\text{subject to} && \sum_{i} x_k^{(i)} \geq y_k && \forall k && \leftarrow \text{demand} \\
& && \sum_{k} g_k^{(i)}(x_k^{(i)}) \leq c^{(i)} && \forall i && \leftarrow \text{resource capacity}
\end{aligned}$$

Assumption

$f_k^{(i)}(\cdot)$, $g_k^{(i)}(\cdot)$, are convex and quadratic:

$$f_k^{(i)}(x) = \alpha_{k,2}^{(i)} x^2 + \alpha_{k,1}^{(i)} x + \alpha_{k,0}^{(i)}$$

$$g_k^{(i)}(x) = \beta_{k,2}^{(i)} x^2 + \beta_{k,1}^{(i)} x + \beta_{k,0}^{(i)}$$

Resource allocation problem definition

If demand sequence: $\{y_1, \ldots, y_n\}$ is known

Optimal allocation for no uncertainty

$$\underset{x_k^{(i)} \in [\underline{x}_k^{(i)}, \bar{x}_k^{(i)}]}{\text{minimize}} \quad \sum_{i,k} f_k^{(i)}(x_k^{(i)}) \qquad \leftarrow \text{total cost}$$

$$\text{subject to} \quad \sum_i x_k^{(i)} \geq y_k \qquad \forall k \qquad \leftarrow \text{demand}$$

$$\sum_k g_k^{(i)}(x_k^{(i)}) \leq c^{(i)} \qquad \forall i \qquad \leftarrow \text{resource capacity}$$

Assumption

$f_k^{(i)}(\cdot)$, $g_k^{(i)}(\cdot)$, are convex and quadratic:

$$f_k^{(i)}(x) = \alpha_{k,2}^{(i)} x^2 + \alpha_{k,1}^{(i)} x + \alpha_{k,0}^{(i)}$$

$$g_k^{(i)}(x) = \beta_{k,2}^{(i)} x^2 + \beta_{k,1}^{(i)} x + \beta_{k,0}^{(i)}$$

Resource allocation problem definition

Unknown demand sequence with samples $y^{(j)} = \{y_1^{(j)}, \ldots, y_n^{(j)}\}$, $j = 1, \ldots, q$

Robust optimal allocation

$$\underset{\substack{x_k^{(i,j)} \in [\underline{x}_k^{(i)}, \bar{x}_k^{(i)}] \\ x_1^{(i)}}}{\text{minimize}} \quad \frac{1}{q} \sum_{i,j,k} f_k^{(i,j)}(x_k^{(i,j)})$$

$$\text{subject to} \quad \sum_i x_k^{(i,j)} \geq y_k^{(j)} \qquad \forall k, j$$

$$\sum_k g_k^{(i,j)}(x_k^{(i,j)}) \leq c^{(i)} \qquad \forall i, j$$

$$x_1^{(i,j)} = x_1^{(i)} \qquad \forall i, j \qquad \leftarrow \text{common 1st decision}$$

Assumption

Samples $y^{(1)}, \ldots, y^{(q)}$ are i.i.d.

Resource allocation problem definition

Unknown demand sequence with samples $y^{(j)} = \{y_1^{(j)}, \ldots, y_n^{(j)}\}$, $j = 1, \ldots, q$

Robust optimal allocation

$$\underset{\substack{x_k^{(i,j)} \in [\underline{x}_k^{(i)}, \bar{x}_k^{(i)}] \\ x_1^{(i)}}}{\text{minimize}} \quad \frac{1}{q} \sum_{i,j,k} f_k^{(i,j)}(x_k^{(i,j)})$$

$$\text{subject to} \quad \sum_i x_k^{(i,j)} \geq y_k^{(j)} \qquad \forall k, j$$

$$\sum_k g_k^{(i,j)}(x_k^{(i,j)}) \leq c^{(i)} \qquad \forall i, j$$

$$x_1^{(i,j)} = x_1^{(i)} \qquad \forall i, j \qquad \leftarrow \text{common 1st decision}$$

Assumption

Samples $y^{(1)}, \ldots, y^{(q)}$ are i.i.d.

Algorithm

Equivalent problem:

$$\begin{aligned}
\underset{\substack{x_k^{(i,j)},\ x_1^{(i)} \\ z_k^{(i,j)},\ s_k^{(j)},\ t^{(i,j)}}}{\text{minimize}} \quad & \sum_{i,j,k} \left[\frac{1}{q} f_k^{(i,j)}(x_k^{(i,j)}) + \mathcal{I}_{[\underline{x}_k^{(i)},\, \bar{x}_k^{(i)}]}(x_k^{(i,j)}) \right] \\
& + \sum_{j,k} \mathcal{I}_{\geq 0}(s_k^{(j)}) + \sum_{i,j} \mathcal{I}_{\leq c^{(i)}}(t^{(i,j)}) \\
\text{subject to} \quad & g_k^{(i,j)}(x_k^{(i,j)}) = z_k^{(i,j)} \qquad \forall i, j, k \\
& \sum_i x_k^{(i,j)} - y_k^{(j)} = s_k^{(j)} \qquad \forall j, k \\
& \sum_k z_k^{(i,j)} = t^{(i,j)} \qquad \forall i, j \\
& x_1^{(i,j)} = x_1^{(i)} \qquad \forall i, j
\end{aligned}$$

Algorithm

Equivalent problem:

$$\begin{aligned}
\underset{\substack{x_k^{(i,j)},\ x_1^{(i)} \\ z_k^{(i,j)},\ s_k^{(j)},\ t^{(i,j)}}}{\text{minimize}} \quad & \sum_{i,j,k} \left[\tfrac{1}{q} f_k^{(i,j)}(x_k^{(i,j)}) + \mathcal{I}_{[\underline{x}_k^{(i)},\, \bar{x}_k^{(i)}]}(x_k^{(i,j)}) \right] \\
& + \sum_{j,k} \mathcal{I}_{\geq 0}(s_k^{(j)}) + \sum_{i,j} \mathcal{I}_{\leq c^{(i)}}(t^{(i,j)}) \\
\text{subject to} \quad & g_k^{(i,j)}(x_k^{(i,j)}) = z_k^{(i,j)} \qquad \forall i,j,k \\
& \sum_i x_k^{(i,j)} - y_k^{(j)} = s_k^{(j)} \qquad \forall j,k \\
& \sum_k z_k^{(i,j)} = t^{(i,j)} \qquad \forall i,j \\
& x_1^{(i,j)} = x_1^{(i)} \qquad \forall i,j
\end{aligned}$$

$$\mathcal{I}_{\mathcal{S}}(x) = \begin{cases} 0 & x \in \mathcal{S} \\ +\infty & x \notin \mathcal{S} \end{cases}$$

Algorithm

Equivalent problem:

$$\underset{\substack{x_k^{(i,j)},\ x_1^{(i)} \\ z_k^{(i,j)},\ s_k^{(j)},\ t^{(i,j)}}}{\text{minimize}} \quad \sum_{i,j,k} \left[\frac{1}{q} f_k^{(i,j)}(x_k^{(i,j)}) + \mathcal{I}_{[\underline{x}_k^{(i)},\, \bar{x}_k^{(i)}]}(x_k^{(i,j)}) \right]$$

$$+ \sum_{j,k} \mathcal{I}_{\geq 0}(s_k^{(j)}) + \sum_{i,j} \mathcal{I}_{\leq c^{(i)}}(t^{(i,j)})$$

$$\text{subject to} \quad g_k^{(i,j)}(x_k^{(i,j)}) = z_k^{(i,j)} \qquad \forall i, j, k$$

$$\sum_i x_k^{(i,j)} - y_k^{(j)} = s_k^{(j)} \qquad \forall j, k$$

$$\sum_k z_k^{(i,j)} = t^{(i,j)} \qquad \forall i, j$$

$$x_1^{(i,j)} = x_1^{(i)} \qquad \forall i, j$$

split capacity constraints into: separable nonlinearities & linear constraints

Algorithm

Augmented Lagrangian:

$$
\begin{aligned}
L = & \sum_{i,j,k} \tfrac{1}{q} f_k^{(i,j)}(x_k^{(i,j)}) \\
& + \sum_{i,j,k} \left[\mathcal{I}_{[\underline{x}_k^{(i)}, \bar{x}_k^{(i)}]}(x_k^{(i,j)}) + \tfrac{\rho_1}{2} \left(z_k^{(i,j)} - g_k^{(i,j)}(x_k^{(i,j)}) + \lambda_k^{(i,j)} \right)^2 \right] \\
& + \sum_{i,j} \left[\mathcal{I}_{\leq c^{(i)}}(t^{(i,j)}) + \tfrac{\rho_2}{2} \left(t^{(i,j)} - \sum_k z_k^{(i,j)} + p^{(i,j)} \right)^2 \right] \\
& + \sum_{j,k} \left[\mathcal{I}_{\geq 0}(s_k^{(j)}) + \tfrac{\rho_3}{2} \left(s_k^{(j)} - \sum_i x_k^{(i,j)} + y_k^{(j)} + \mu_k^{(j)} \right)^2 \right] \\
& + \sum_{i,j} \tfrac{\rho_4}{2} \left(x_1^{(i)} - x_1^{(i,j)} + \nu^{(i,j)} \right)^2
\end{aligned}
$$

- ⋆ $\lambda_k^{(i,j)}$, $\mu_k^{(j)}$, $\nu^{(i,j)}$, $p^{(i,j)}$: Lagrange multipliers
- ⋆ ρ_1, ρ_2, ρ_3: multiplier update step size parameters

Algorithm

ADMM iteration: primal update

$$x_k^{(i,j)} \leftarrow \underset{x_k^{(i,j)}}{\operatorname{argmin}} L = \Pi_{[\underline{x}_k^{(i)}, \bar{x}_k^{(i)}]}\{\text{minimizer of quartic equation in } x_k^{(i,j)}\}$$

$$z_k^{(i,j)} \leftarrow \underset{z_k^{(i,j)}}{\operatorname{argmin}} L =$$

$$g_k^{(i,j)}(x_k^{(i,j)}) - \lambda_k^{(i,j)} + \frac{\rho_2}{\rho_1 + n\rho_2}\left[t^{(i,j)} + p^{(i,j)} - \sum_k \left(g_k^{(i,j)}(x_k^{(i,j)}) - \lambda_k^{(i,j)}\right)\right]$$

$$x_1^{(i)} \leftarrow \underset{x_1^{(i)}}{\operatorname{argmin}} L = \sum_j (x_1^{(i,j)} - \nu^{(i,j)})$$

$$t^{(i,j)} \leftarrow \underset{t^{(i,j)}}{\operatorname{argmin}} L = \Pi_{\leq c^{(i)}}\left\{\sum_k z_k^{(i,j)} - p^{(i,j)}\right\}$$

$$s_k^{(j)} \leftarrow \underset{s_k^{(j)}}{\operatorname{argmin}} L = \Pi_{\geq 0}\left\{\sum_i x_k^{(i,j)} - y_k^{(j)} - \mu_k^{(j)}\right\}$$

Algorithm

ADMM iteration: primal update

$$x_k^{(i,j)} \leftarrow \underset{x_k^{(i,j)}}{\operatorname{argmin}} L = \Pi_{[\underline{x}_k^{(i)}, \bar{x}_k^{(i)}]}\{\text{minimizer of quartic equation in } x_k^{(i,j)}\}$$

$$z_k^{(i,j)} \leftarrow \underset{z_k^{(i,j)}}{\operatorname{argmin}} L =$$

$$g_k^{(i,j)}(x_k^{(i,j)}) - \lambda_k^{(i,j)} + \frac{\rho_2}{\rho_1 + n\rho_2}\left[t^{(i,j)} + p^{(i,j)} - \sum_k \left(g_k^{(i,j)}(x_k^{(i,j)}) - \lambda_k^{(i,j)}\right)\right]$$

$$x_1^{(i)} \leftarrow \underset{x_1^{(i)}}{\operatorname{argmin}} L = \sum_j (x_1^{(i,j)} - \nu^{(i,j)})$$

$$t^{(i,j)} \leftarrow \underset{t^{(i,j)}}{\operatorname{argmin}} L = \Pi_{\leq c^{(i)}}\left\{\sum_k z_k^{(i,j)} - p^{(i,j)}\right\}$$

$$s_k^{(j)} \leftarrow \underset{s_k^{(j)}}{\operatorname{argmin}} L = \Pi_{\geq 0}\left\{\sum_i x_k^{(i,j)} - y_k^{(j)} - \mu_k^{(j)}\right\}$$

can be implemented in parallel

partially parallelizable

Algorithm

ADMM iteration: dual update

$$\mu_k^{(j)} \leftarrow \mu_k^{(j)} + s_k^{(j)} - \sum_i x_k^{(i,j)} + y_k^{(j)}$$

$$\lambda_k^{(i,j)} \leftarrow \lambda_k^{(i,j)} + z_k^{(i,j)} - g_k^{(i,j)}(x_k^{(i,j)})$$

$$\nu^{(i,j)} \leftarrow \nu^{(i,j)} + x_1^{(i)} - x_1^{(i,j)}$$

$$p^{(i,j)} \leftarrow p^{(i,j)} + t^{(i,j)} - \sum_k z_k^{(i,j)}$$

Algorithm

ADMM iteration: dual update

$$\mu_k^{(j)} \leftarrow \mu_k^{(j)} + s_k^{(j)} - \sum_i x_k^{(i,j)} + y_k^{(j)}$$

$$\lambda_k^{(i,j)} \leftarrow \lambda_k^{(i,j)} + z_k^{(i,j)} - g_k^{(i,j)}(x_k^{(i,j)})$$

$$\nu^{(i,j)} \leftarrow \nu^{(i,j)} + x_1^{(i)} - x_1^{(i,j)}$$

$$p^{(i,j)} \leftarrow p^{(i,j)} + t^{(i,j)} - \sum_k z_k^{(i,j)}$$

can be implemented in parallel

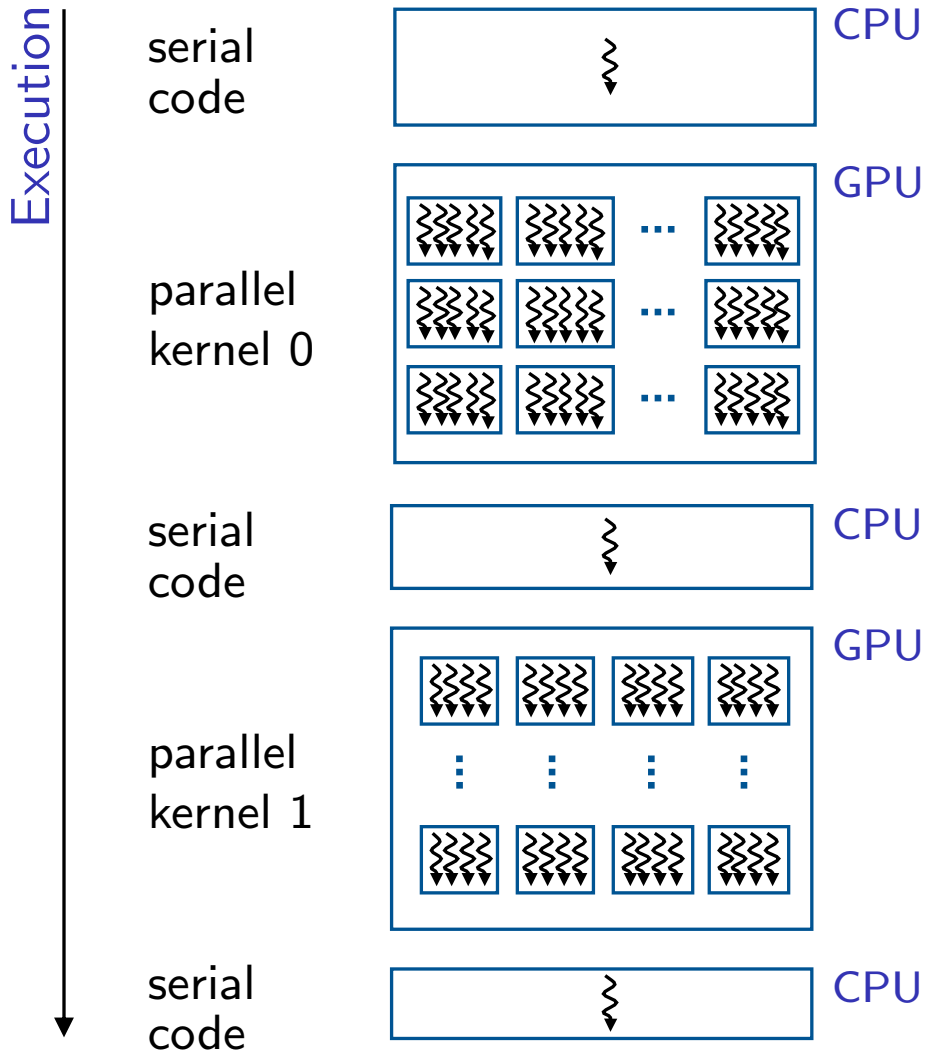
partially parallelizable

Parallel GPU implementation

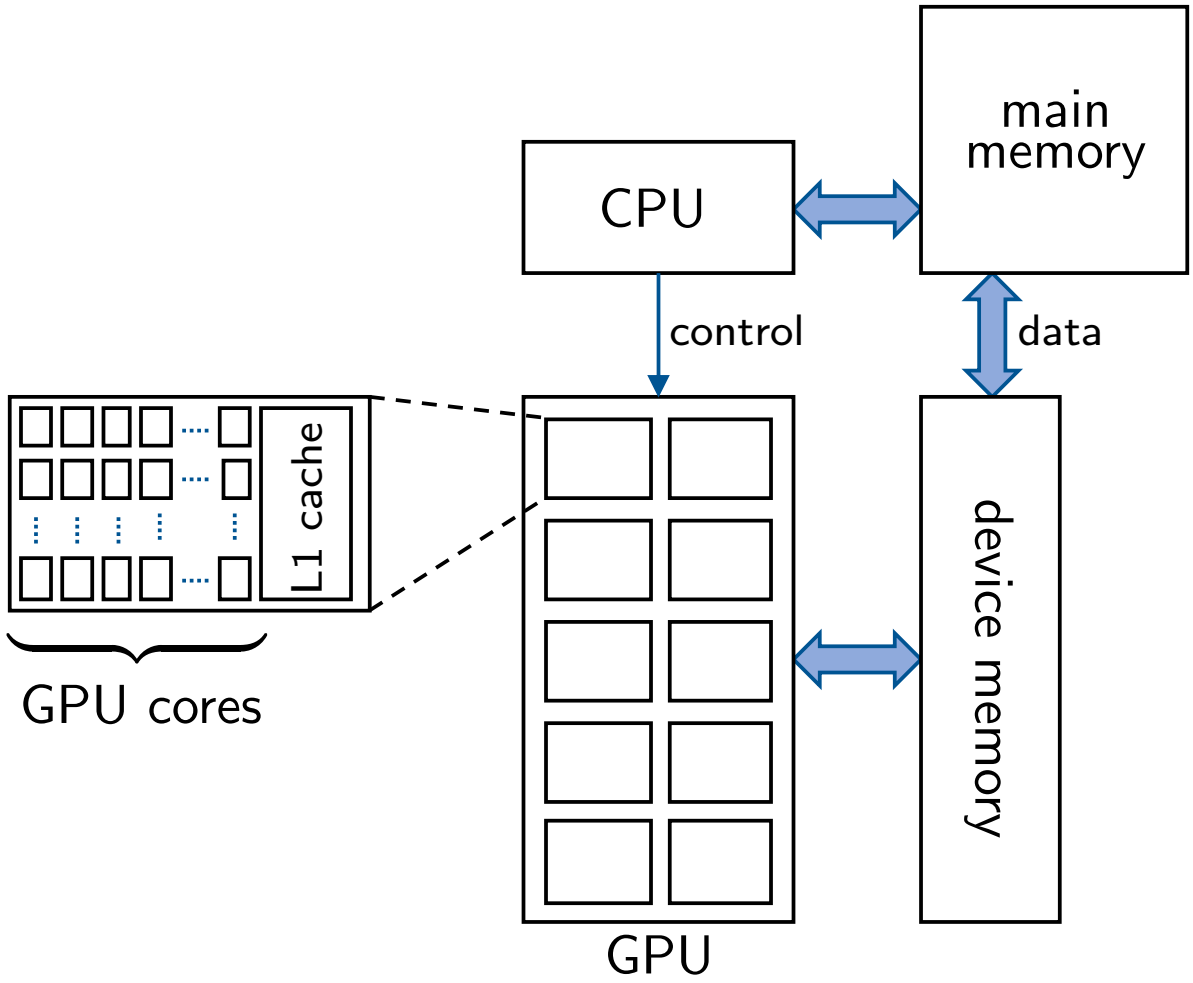
CUDA heterogeneous programming model

CUDA kernels run in parallel on the GPU

Threads execute same instructions simultaneously using different data



Parallel GPU implementation



e.g. Nvidia GTX 1060 3GB GPU has 1152 cores and up to 18432 threads

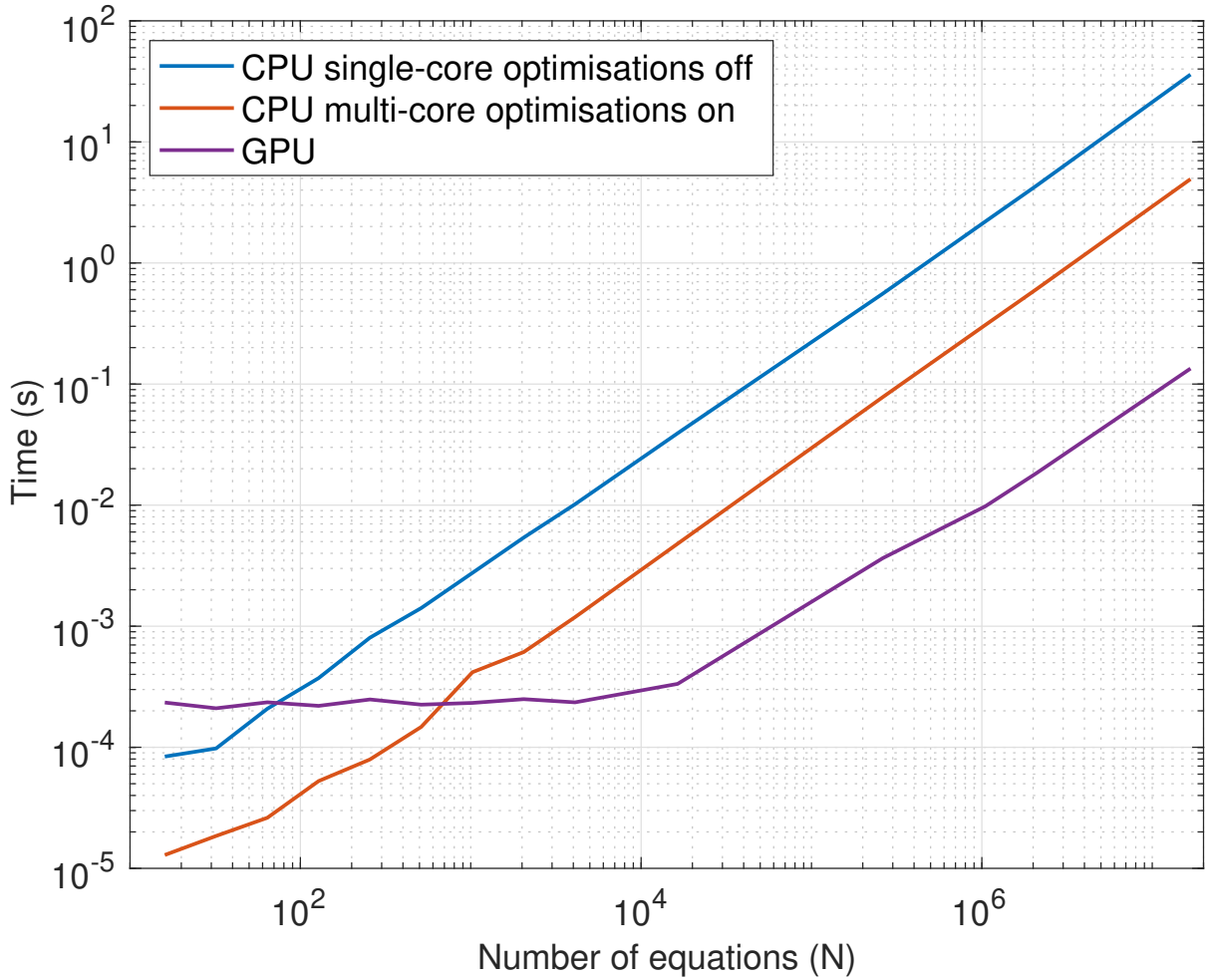
Quartic minimization

- $x^* = \operatorname{argmin}_x Ax^4 + Bx^3 + Cx^2 + Dx$
- Fast algebraic solution based on Vieta's and Cardano's methods:

Input : coefficients A, B, C, D
$b \leftarrow 3B/4A$, $c \leftarrow C/2A$, $d \leftarrow D/4A$;
$Q \leftarrow c/3 - b^2/9$, $R \leftarrow bc/6 - b^3/27 - d/2$, $\Delta \leftarrow Q^3 + R^2$;
if $\Delta > 0$ **then**
  $x^* \leftarrow (R + \sqrt{\Delta})^{1/3} + (R - \sqrt{\Delta})^{1/3} - b/3$;
else if $Q = R = 0$ **then**
  $x^* \leftarrow -b/3$;
else
  $\theta \leftarrow \cos^{-1}(R/|Q|^{3/2})$;
  $x_a \leftarrow 2|Q|^{1/2}\cos(\theta/3) - b/3$;
  $x_b \leftarrow 2|Q|^{1/2}\cos(\theta/3 + 2\pi/3) - b/3$;
  $x_c \leftarrow 2|Q|^{1/2}\cos(\theta/3 + 4\pi/3) - b/3$;
  $(x_1, x_2, x_3) \leftarrow \mathsf{sort}(x_a, x_b, x_c)$;
  $\delta f \leftarrow \frac{1}{4}(x_1^4 - x_3^4) + \frac{b}{3}(x_1^3 - x_3^3) + \frac{c}{2}(x_1^2 - x_3^2) + d(x_1 - x_3)$;
  if $\delta f > 0$ **then**
    $x^* \leftarrow x_3$;
  else
    $x^* \leftarrow x_1$;

Quartic minimization

Computation time for minimization of N quartics with random coefficients
(CPU speed optimizations via compiler flags /Ox and /Od)



Application: PHEV energy management

- $\star$ $y_k^{(j)}$: samples of stochastic driver power demand
- $\star$ $x_k^{(1,j)}$: drive power from i.c. engine
- $\star$ $x_k^{(2,j)}$: drive power from electric motor
- $\star$ Objective: minimize fuel consumption ($f_k^{(2,j)} = 0 \ \forall j,k$)
  constraints: battery capacity & power flows ($g_k^{(1,j)} = 0 \ \forall j,k$)

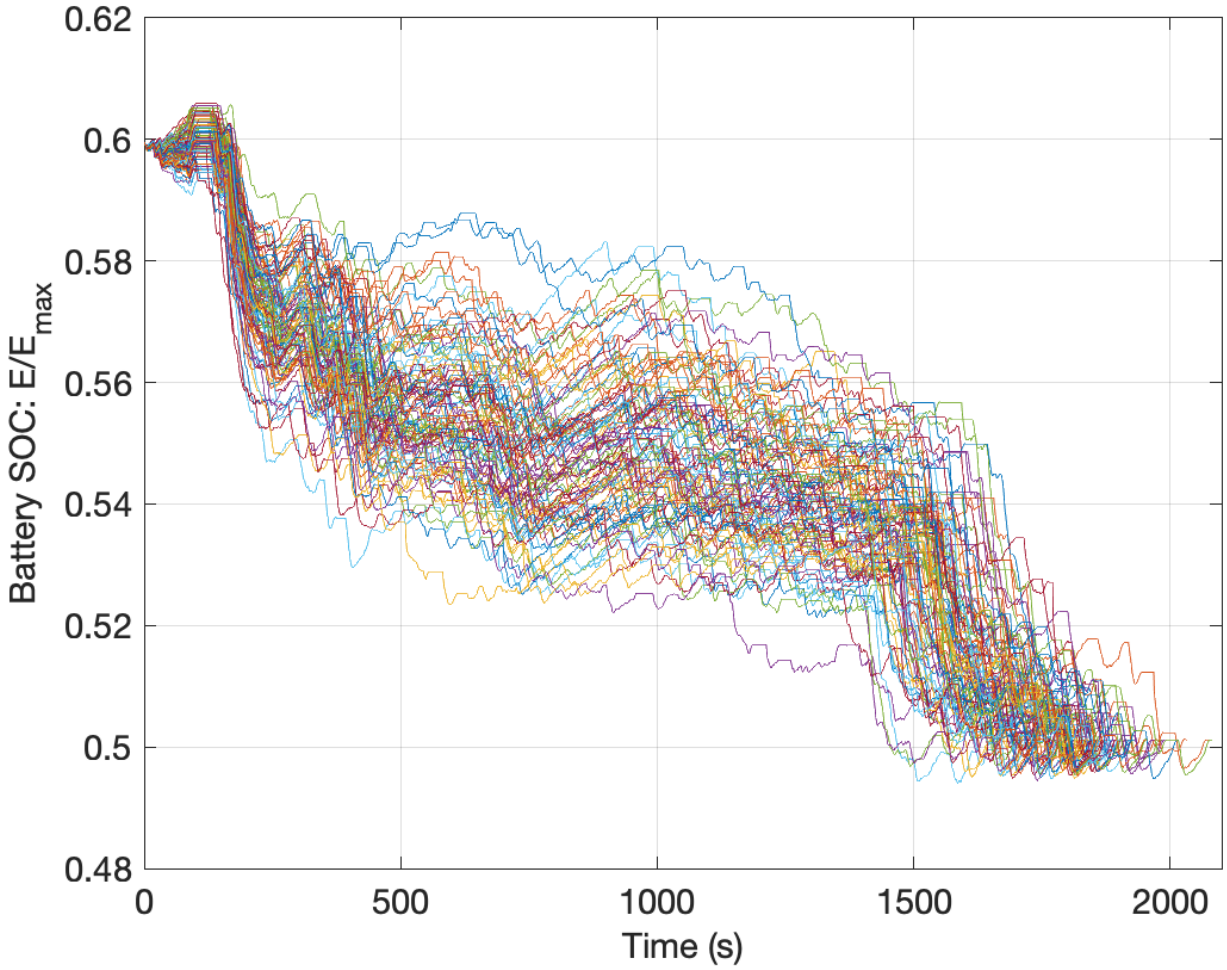
Robust optimization problem

$$\underset{\substack{x_k^{(i,j)} \in [\underline{x}_k^{(i)}, \bar{x}_k^{(i)}] \\ x_1^{(i)}}}{\text{minimize}} \quad \frac{1}{q} \sum_{j=1}^{q} \sum_{k=1}^{n} f_k^{(1,j)}(x_k^{(1,j)})$$

$$\begin{aligned}
\text{subject to} \quad & x_k^{(1,j)} + x_k^{(2,j)} \geq y_k^{(j)}, && j \in \{1,\dots,q\},\ k \in \{1,\dots n\} \\
& \textstyle\sum_k g_k^{(2,j)}(x_k^{(2,j)}) \leq \Delta E, && j \in \{1,\dots,q\} \\
& x_1^{(i,j)} = x_1^{(i)}, && i = 1,2,\ j \in \{1,\dots,q\}
\end{aligned}$$

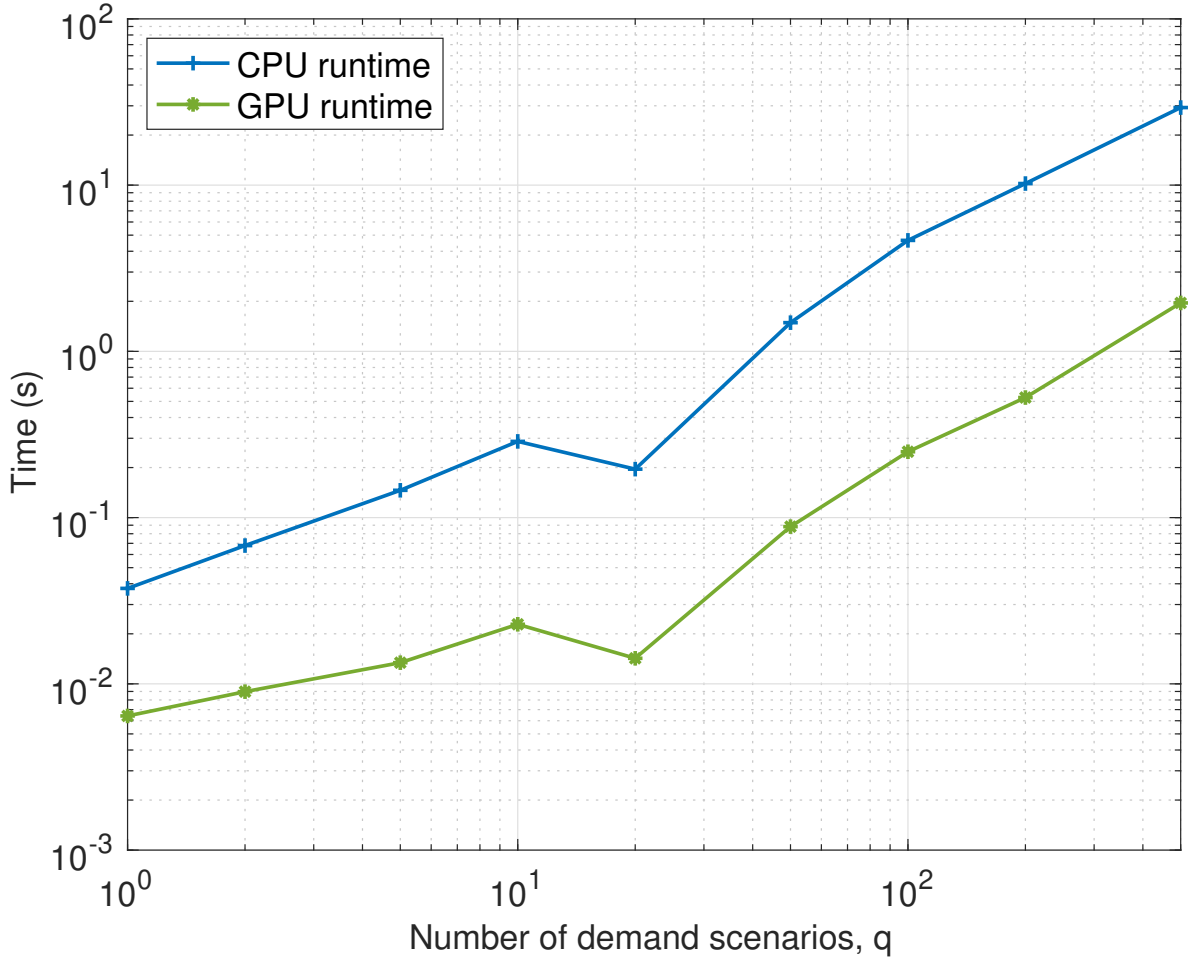
Application: PHEV energy management

Optimal predicted battery state of charge profiles with 100 power demand scenarios generated from random perturbations of FTP-75 drive cycle



Application: PHEV energy management

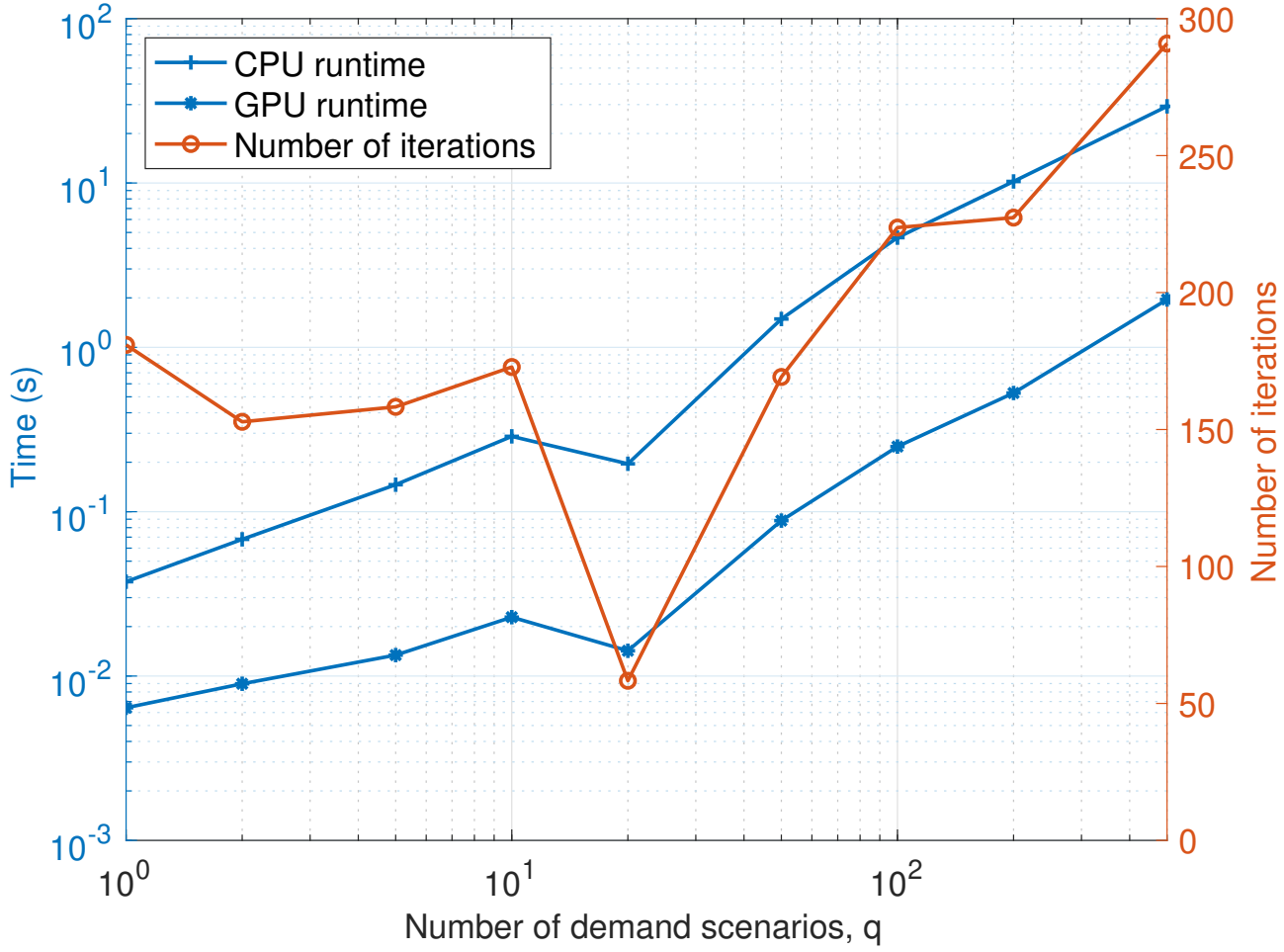
Average computation times



- ⋆ Parallel implementation is between $10\times$ and $20\times$ faster than serial
- ⋆ 0.25 s for 100 scenarios (0.6 s for 200 scenarios) is acceptable with $T_{\text{samp}} = 1\,\text{s}$

Application: PHEV energy management

Average computation times



- ⋆ Parallel implementation is between $10\times$ and $20\times$ faster than serial
- ⋆ 0.25 s for 100 scenarios (0.6 s for 200 scenarios) is acceptable with $T_{\text{samp}} = 1\,\text{s}$

Conclusions

Contributions

- ADMM algorithm for robust quadratic resource allocation problems
- Parallel implementation on GPU coded in CUDA

Observations

- Choice of operator splitting is important for parallel implementation
- Robust optimal energy management problem is solvable in real time using cheap, non-specialized hardware
- State of the art low-cost parallel processing hardware is evolving fast

Code: https://github.com/qureshizawar/CUDA-quartic-solver

Questions?

Conclusions

Contributions

- ADMM algorithm for robust quadratic resource allocation problems
- Parallel implementation on GPU coded in CUDA

Observations

- Choice of operator splitting is important for parallel implementation
- Robust optimal energy management problem is solvable in real time using cheap, non-specialized hardware
- State of the art low-cost parallel processing hardware is evolving fast

Code: https://github.com/qureshizawar/CUDA-quartic-solver

Questions?

Conclusions

Contributions

- ADMM algorithm for robust quadratic resource allocation problems
- Parallel implementation on GPU coded in CUDA

Observations

- Choice of operator splitting is important for parallel implementation
- Robust optimal energy management problem is solvable in real time using cheap, non-specialized hardware
- State of the art low-cost parallel processing hardware is evolving fast

Code: https://github.com/qureshizawar/CUDA-quartic-solver

Questions?

Questions?