Computational Approaches for Efficient Scheduling of Steel Plants as Demand Response Resource

Xiao Zhang[1] Gabriela Hug[2] J. Zico Kolter[1] Iiro Harjunkoski[3]

[1]Carnegie Mellon University

[2]ETH Zurich

[3]ABB Corporate Research

PSCC 2016, Geona, Italy

Demand Response

- The goal: sustainable energy future and a green planet
  - renewable generation: wind turbines, solar panels, etc.
  - however, power output uncertain
  - need more balancing power
- Power balance
  - generation equals demand
  - traditional balancing power: generators
  - generators frequent adjustment, not economical
- Demand response
  - adjust the other side of the equation
  - potentially provides a cost-effective solution

Industrial Loads

Demand response resource (DRR)

- residential, commercial, industrial loads
- e.g. residential areas, electric vehicles, buildings, data centers, pumps, furnaces, fans, aluminum smelters, cement crushers, ...

Industrial load as DRR

- Advantages
  - infrastructure
    - already installed
  - response
    - large, fast, accurate
  - economic incentive
    - strong
- Challenges
  - reliability
    - critical safety constraint
  - complexity
    - production activities
  - granularity
    - power change response

Outline

1. Steel Plant as Demand Response Resource
   - Steel Plant Scheduling
   - Mathematical Model
2. Computation Methods
   - Additional Constraints as Cuts
   - Tailored branch and bound algorithm
3. Numerical Studies
4. Summary

Steel Manufacturing

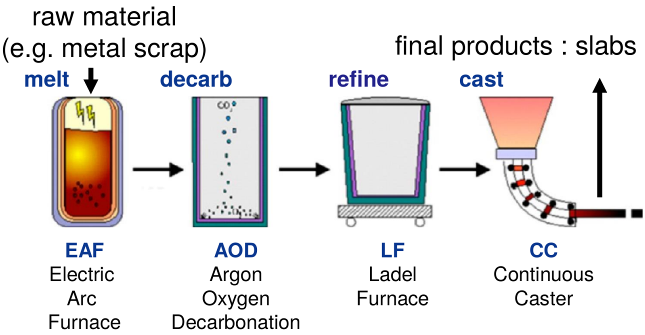


Figure: Production process of steel manufacturing

Heat: a certain amount of metal (batch)
- quantify the production throughput

Steel Plant Scheduling

One of the most difficult industrial processes for scheduling

- large-scale, multi-product, multi-stage, parallel equipment, critical production-related constraints, etc.
- thousands of binary variables

Energy intensive

- energy cost is significant
- great potential as demand response resource

Scheduling goal

- traditionally, minimize the make-span
- we consider daily scheduling and minimize its daily cost in electricity energy market

Resource Task Network (RTN)

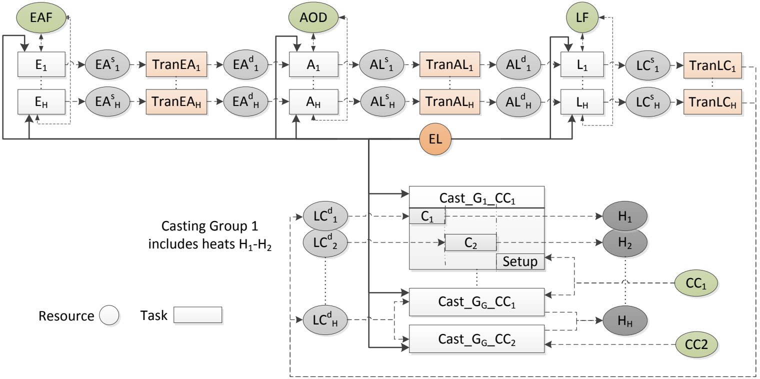


Figure: Resource task network model for a steel plant

Mathematical Formulations

Constraints

- resource balance

$$y_{s,t} = y_{s,t-1} + \sum_{k \in \mathbb{K}} \sum_{\theta=0}^{\tau_k} \Delta_{s,\theta}^{k} \cdot x_{k,t-\theta} \quad \forall s \in \mathbb{S}_{\neg\{\mathrm{EL}\}}, \forall t$$

$$y_{\mathrm{EL},t} = \sum_{k \in \mathbb{K}} \sum_{\theta=0}^{\tau_k} \Delta_{\mathrm{EL},\theta}^{k} \cdot x_{k,t-\theta} \qquad \forall t$$

- task execution
- waiting time
- product delivery

Objective

- minimize electricity cost

Additional constraints as cuts

In steel manufacturing

- many tasks are equivalent to each other
  - e.g. the decarburization of molten metal for two similar batches of products
- the casting sequence for heats belonging to the same casting group are pre-specified
  - e.g. from expert experiences or casting optimization
- impose an enforced processing order
  - thereby, reduce the search space of the MIP problem

Additional cuts

$$\sum_{t' \leq t} (x_{k_1,t'} - x_{k_2,t'}) \geq 0 \quad \forall t, (k_1, k_2) \in \mathbb{O}$$

Tailored Branch and Bound Algorithm

Branch and Bound

- commercial solvers
  - e.g. CPLEX, Gurobi
  - powerful, but are designed for general optimization problems
- tailored by special features
  - the heats belonging to the same campaign group are generally processed close to each other

For each casting group

- *leader* (first heat) and *followers* (other heats)
- require the leader to be processed first
- require its followers to be processed within certain time ranges
  - pre-calculated time ranges, before optimization

Tailored Branch and Bound Algorithm

```
procedure TailoredBranchBound
    q ← Priority-Queue()                         ▷ pops largest objective first
    q2 ← Priority-Queue()                        ▷ pops smallest objective first
    q.push(SolveRelaxation({ }))
    q2.push(FindIntegerSolutionHeuristics())
    while q not empty do
        (f, x, y, C) ← q.pop()
        q2.push(Rounding((f, x, y, C)))
        if q2.first - f ≤ ϵ then
            return q2.pop()
        else
            for C_i in BranchNodes(C) do
                q.push(SolveRelaxation(C_i))
            end for
        end if
    end while
end procedure
```

Figure 4. Tailored branch and bound algorithm

Tailored Branch and Bound Algorithm

function BranchNodes(C)
  if C == { } **then**
    return $[(0, T), \ldots, (0, T)]$
  else
    $k^* \leftarrow \arg\max_{k \in L.\text{keys}}(b_k - a_k)$
    if $b_{k^*} - a_{k^*} > \epsilon_d$ **then**
      $m^* \leftarrow \text{int}(\frac{b_{k^*} - a_{k^*}}{2})$
      $\{k : (d_a, d_b)\} \leftarrow L[k^*]$
      $C_1 \leftarrow [\ldots, (a_{k^*}, m^*), (a_{k^*} + d_a, m^* + d_b), \ldots]$
      $C_2 \leftarrow [\ldots, (m^*, b_{k^*}), (m^* + d_a, b_{k^*} + d_b), \ldots]$
      return $\{C_1, C_2\}$
    else
      $k^* \leftarrow \arg\max_{k \in \mathbb{K}}(b_k - a_k)$
      $m^* \leftarrow \text{int}(\frac{b_{k^*} - a_{k^*}}{2})$
      $C_1 \leftarrow [\ldots, (a_{k^*}, m^*), \ldots]$
      $C_2 \leftarrow [\ldots, (m^*, b_{k^*}), \ldots]$
      return $\{C_1, C_2\}$
    end if
  end if
end function

Figure 5. Branch by leader heats

Steel Plant Parameters

Table: Nominal power consumptions [MW]

equipment	EAF_1	EAF_2	AOD_1	AOD_2	LF_1	LF_2	CC_1	CC_2
power	85	85	2	2	2	2	7	7

Table: Steel heat/group correspondence

group	G_1	G_2	G_3	G_4	G_5	G_6
heats	$H_1 - H_4$	$H_5 - H_8$	$H_9 - H_{12}$	$H_{13} - H_{17}$	$H_{18} - H_{20}$	$H_{21} - H_{24}$

Table: Nominal processing times [min]

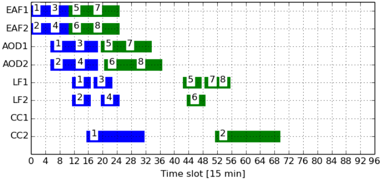
heats	EAF_1	EAF_2	AOD_1	AOD_2	LF_1	LF_2	CC_1	CC_2
$H_1 - H_4$	80	80	75	75	35	35	50	50
$H_5 - H_6$	85	85	80	80	40	40	60	60
$H_7 - H_8$	85	85	80	80	20	20	55	55
$H_9 - H_{12}$	90	90	95	95	45	45	60	60
$H_{13} - H_{14}$	85	85	85	85	25	25	70	70
$H_{15} - H_{16}$	85	85	85	85	25	25	75	75
H_{17}	80	80	85	85	25	25	75	75
H_{18}	80	80	95	95	45	45	60	60
H_{19}	80	80	95	95	45	45	70	70
H_{20}	80	80	95	95	30	30	70	70
$H_{21} - H_{22}$	80	80	80	80	30	30	50	50
$H_{23} - H_{24}$	80	80	80	80	30	30	50	60

Computational Results

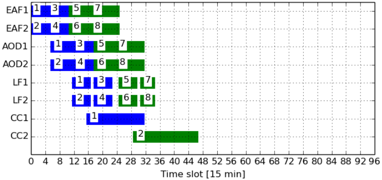
Table: Branch and bound results with $t_0 = 15$min

Groups		c0	c1	b1
G1-2	Obj(k$)	24.553	24.553	24.698
	CPU(s)	5.8	3.7	6.2
	lpNum	2460	1985	57
G1-3	Obj(k$)	39.306	39.308	39.665
	CPU(s)	155.4	60.7	50.0
	lpNum	9071	3835	228
G1-4	Obj(k$)	57.857	57.857	58.694
	CPU(s)	60.4	42.7	197.8
	lpNum	3852	2745	280
G1-6	Obj(k$)	86.352	86.352	86.799
	CPU(s)	104.9	80.4	2737.6
	lpNum	3698	2631	725

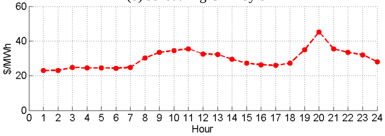
Scheduling Comparison



(a) scheduling G1-2 by c0



(b) scheduling G1-2 by b1



(c) hourly energy prices

B&B Iterations

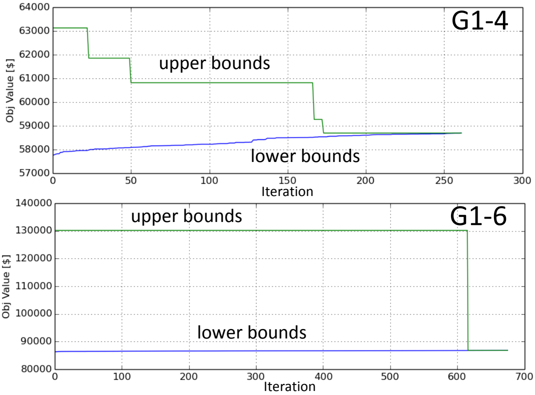


Figure: Branch and bound iterations

Summary

- The proposed methods show potentials to make the computations more tractable
  - cuts to reduce search space
  - tailored b&b algorithm
  - cpu time, iteration number
- Make it more appealing for industrial plants such as steel plants to take part in demand response
- Outlook
  - find a better rounding method
  - more accurate modeling of steel plants
  - etc.

ABB Power and productivity for a better world™

Carnegie Mellon University

Thanks!

contact: xiaozhang@cmu.edu