Graduate Program Numerics

Optimization and optimal decision

M. Debouté[1] L. Hirwa-Shema[1] A. Barrabès[2]

[1]M2 CMI Optim
Université de Bordeaux

[2]M2 Economic risks and data science
Université de Bordeaux

Friday, 9th September 2022

université de BORDEAUX

Google ROADEF/EURO challenge

Large-scale machine reassignment problem : improving the utilization of a set of machines.

- Processes assigned to machines consuming ressources.
- Hard constraints on moving processes.
- Several costs : load, balance and move related costs.

Google ROADEF/EURO challenge

Large-scale machine reassignment problem : improving the utilization of a set of machines.

- Processes assigned to machines consuming ressources.
- Hard constraints on moving processes.
- Several costs : load, balance and move related costs.

Objective

The objective consists in minimizing the weighted sum of all previous costs.

Variables

Let $p \in \mathcal{P}, m \in \mathcal{M}, s \in \mathcal{S}, l \in \mathcal{L}, n \in \mathcal{N}, r \in \mathcal{R}$ and $b \in \mathcal{B}$.

- $x_{p,m} = \begin{cases} 1 & \text{if } M(p) = m \\ 0 & \text{otherwise} \end{cases}.$
- $y_{s,l} = \begin{cases} 1 & \text{if } \exists (p,m) \in (s \times l) \mid M(p) = m \\ 0 & \text{otherwise} \end{cases}.$
- $z_{s,n} = \begin{cases} 1 & \text{if } \exists (p,m) \in (s \times n) \mid M(p) = m \\ 0 & \text{otherwise} \end{cases}.$
- $lc_{m,r} \geqslant 0, lc_r \geqslant 0, a_{m,r} \geqslant 0, bc_{b,m} \geqslant 0$ and $bc_b \geqslant 0.$

Constraints

- $\forall p \in \mathcal{P}, \sum_{m \in \mathcal{M}} x_{p,m} = 1.$ (Assignment)
- $\forall (r,m) \in (\mathcal{R} \times \mathcal{M}), \sum_{p \in \mathcal{P}} x_{p,m} \times R_{p,r} \leqslant C_{m,r}.$ (Hard capacity)
- $\forall (s,m) \in (\mathcal{S} \times \mathcal{M}), \sum_{p \in s} x_{p,m} \leqslant 1.$ (Conflict)
- Let $s \in \mathcal{S}$ and $l \in \mathcal{L}$. $\forall (p,m) \in (s \times l), x_{p,m} \leqslant y_{s,l}.$ (Spread 1)
- $\forall (l,s) \in (\mathcal{L} \times \mathcal{S}), \sum_{p \in s} (\sum_{m \in l} x_{p,m}) \geqslant y_{s,l}.$ (Spread 2)
- $\forall s \in \mathcal{S}, \sum_{l \in \mathcal{L}} y_{s,l} \geqslant spreadMin(s).$ (Spread 3)

Constraints

- $\forall s_1 \in \mathcal{S}, \forall s_2 \in Dep(s_1), \forall n \in \mathcal{N}, z_{s_1,n} \leqslant z_{s_2,n}.$ (Dependency 1)
- $\forall s \in \mathcal{S}, \forall n \in \mathcal{N}, \forall p \in \mathcal{S}, \forall m \in n, x_{p,m} \leqslant z_{s,n}.$ (Dependency 2)
- $\forall s \in \mathcal{S}, \forall n \in \mathcal{N}, \sum\limits_{m \in n} \big(\sum\limits_{p \in s} x_{p,m} \big) \geqslant z_{s,n}.$ (Dependency 3)
- $\forall r \in \mathrm{TR}, \forall m \in \mathcal{M},$
  $\sum\limits_{p | M_0(p) = m} R_{p,r} + \sum\limits_{p | M_0(p) \neq m} R_{p,m} \times x_{p,m} \leqslant C_{m,r}.$ (Transient)

Constraints

- $\forall (r,m) \in (\mathcal{R} \times \mathcal{M}), \sum_{p \in \mathcal{P}} (x_{p,m} \times R_{p,r}) - SC_{m,r} \leqslant lc_{m,r}$
  and $lc_r := \sum_{m \in \mathcal{M}} lc_{m,r}$. (Load cost)
- $\forall (r,m) \in (\mathcal{R} \times \mathcal{M}), a_{m,r} = C_{m,r} - \sum_{p \in \mathcal{P}} R_{p,r} \times x_{p,m}$
  $\forall (b,m) \in (\mathcal{B} \times \mathcal{M}), bc_{b,m} \geqslant t_b \times a_{m,r_1} - a_{m,r_2}$
  and $\forall b \in \mathcal{B}, bc_b = \sum_{m \in \mathcal{M}} bc_{b,m}$ (Balance cost)
- $pmc := \sum_{p \in \mathcal{P}} (1 - x_{p,M_0(p)}) \times PMC(p)$. (Process move cost)

Constraints

- $\forall s \in \mathcal{S}, smc_s := \sum_{p \in s} (1 - x_{p,M_0(p)})$ and $\forall s \in \mathcal{S}, smc \geqslant smc_s$. (Service move cost)
- $\forall p \in \mathcal{P}, mmc_p := \sum_{m \in \mathcal{M}} x_{p,m} \times MMC_{M_0(p),m}$ and $mmc := \sum_{p \in \mathcal{P}} mmc_p$. (Machine move cost)

Objective function

Function to minimize

$$\min \sum_{r \in \mathcal{R}} weight_{loadCost}(r) \times lc_r$$
$$+ \sum_{b \in \mathcal{B}} weight_{balanceCost}(b) \times bc_b$$
$$+ weight_{processMoveCost} \times pmc$$
$$+ weight_{serviceMoveCost} \times smc$$
$$+ weight_{machineMoveCost} \times mmc.$$

Testing our code on the instances of the dataset A

Instance	Status	Objective value	Gap (%)	Time (s)
1	Optimal	44306501	0	0.068
2	Feasible	777531842	10^{-4}	Time out
3	Optimal	583005719	0	16.328
4	Feasible	262086758	7.50	Time out
5	Optimal	727578311	0	2.230
6	Feasible	8592396	99.99	Time out
7	Feasible	1111598241	50.38	Time out
8	Feasible	1375256887	24.48	Time out
9	No solution	x	x	Time out
10	No solution	x	x	Time out

8 cores of 2.4 GHz, 8go RAM, time limit : 1 hour,
gap tolerance : 10e-8.

1. Challenge ROADEF/EURO
2. Our work
   - Mathematical model
   - Results
3. In sight

- We obtained 6 good results out of 10. An idea to improve those results is to combine MILP solver with some local search and a metaheuristic.

- We propose the following heuristic algorithm (gradient descent) in order to obtain a better initial solution for the MILP solver.

Data: $initialSol$, $Processes$, $Machines$
Result: $solution$
$solution \leftarrow initialSol$;
$currCost \leftarrow cost(initialSol)$;
$prevCost \leftarrow currCost + 1$;
while $currCost - prevCost < 0$ **do**
    $prevCost \leftarrow currCost$;
    for $p \in Processes$ **do**
        for $m \in Machines$ **do**
            if $isFeasible(move(p, m))$ and $moveCost(p, m) < 0$ **then**
                $currCost \leftarrow currCost + moveCost(p, m)$;
                $solution \leftarrow move(p, m)$
            end
        endfor
    endfor
end