Lecture 2. Nonsmooth optimization, Lagrangian relaxation and applications

M. Gaudioso,*

*DIMES-Universitá della Calabria and ICAR-CNR, Rende (CS), Italia

EUROPT Summer School

August 30th– September 1st, 2021

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual
   - Tackling the Lagrangian dual (LD)
   - Lagrangian heuristics
   - An example: The Set Covering problem (SC)
2. Applications
   - Generalized assignment
   - Spanning tree with minimum branch vertices (ST-MBV)
   - Berth allocation problem in maritime terminals (BAP)
   - Cross docking
3. Conclusions
4. References

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual
   - Tackling the Lagrangian dual (LD)
   - Lagrangian heuristics
   - An example: The Set Covering problem (SC)
2. Applications
   - Generalized assignment
   - Spanning tree with minimum branch vertices (ST-MBV)
   - Berth allocation problem in maritime terminals (BAP)
   - Cross docking
3. Conclusions
4. References

Lagrangian relaxation as a general approach

"Lagrangian relaxation is usually considered in the combinatorial optimization community as a mere technique, sometimes useful to compute bounds. It is actually a very general method"
(C. Lemaréchal, The omnipresence of Lagrange, 4OR, 2003)

The relaxation

Definition (Relaxation)

Given problem (P):

$$\min_{x \in X \subset \mathbb{R}^n} f(x)$$

then problem (P_R):

$$\min_{x \in X_R \subset \mathbb{R}^n} h(x)$$

is a *relaxation* of (P), provided that:

$$X \subset X_R;$$
$$h(x) \leq f(x), \quad x \in X.$$

Let x_R^* and x_P^* be (global) minima of (P_R) and (P), respectively, then

$$h(x_R^*) \leq h(x_P^*) \leq f(x_P^*) \Rightarrow h(x_R^*) \text{ is a lower bound for } (P)$$

Lagrangian relaxation

Consider the following ILP problem

$$\begin{cases} z_{IP} = \min & c^T x \\ \text{s. t.} & Ax = b, \\ & Bx = d \\ & x \geq 0, \text{ integer}, \end{cases}$$

with $x, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^m$ and $d \in \mathbb{R}^p$. We assume that the problem is feasible and that the set

$$X = \{ x \in \mathbb{R}^n \mid Bx = d, \ \ x \geq 0, \ \text{integer} \}$$

is finite, that is $X = \{x_1, x_2, \ldots, x_K\}$.
Let $Ax = b$ be the *complicating* constraints.

Lagrangian relaxation is... a relaxation

Definition (Lagrangian relaxation)

$$\begin{cases} z_{LR}(\lambda) = & \min \ c^T x + \lambda^T (b - Ax) \\ \text{s.t.} & x \in X \end{cases}$$

$z_{LR}(\lambda)$ is the *dual* function.

For $\lambda \in \mathbb{R}^m$, let $x(\lambda)$ be the optimal solution:

$$z_{LR}(\lambda) = c^T x(\lambda) + \lambda^T (b - Ax(\lambda)) = \min_{k=1,\dots,K} c^T x_k + \lambda^T (b - Ax_k)$$

Remark

if $x(\lambda)$ feasible for ILP ($Ax(\lambda) = b$), then it is also optimal.

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual

The Lagrangian dual – Nonsmooth fomulation

Look for the best lower bound.

Definition (Lagrangian dual–LD)

$$z_{LD} = \max_{\lambda} \quad z_{LR}(\lambda),$$

that is

$$z_{LD} = \max_{\lambda} \min_{k=1,\dots,K} c^T x_k + \lambda^T (b - A x_k).$$

Need to solve a *nonsmooth* optimization problem, of the (finite) *max min* type, with objective function piecewise affine and concave. It can be rewritten in LP form (in $\mathbb{R}^{n+1}$):

$$\begin{cases} z_{LD} = & \max\limits_{\lambda, v} \quad v \\ \text{s.t.} & v \le c^T x_k + \lambda^T (b - A x_k), \quad k = 1, \dots, K. \end{cases}$$

The Lagrangian dual (LD). The LP formulation

Write the dual

$$\begin{cases} z_{LD} = & \min\limits_{\mu} \quad c^T(\sum\limits_{k=1}^{K} \mu_k x_k) \\ \text{s.t.} & A(\sum\limits_{k=1}^{K} \mu_k x_k) = b \\ & \sum\limits_{k=1}^{K} \mu_k = 1, \quad \mu_k \geq 0,\ k = 1, \ldots, K, \end{cases}$$

or, equivalently,

$$\begin{cases} z_{LD} = & \min \quad c^T x \\ \text{s.t.} & Ax = b \\ & x \in \text{conv}X = \text{conv}\{\, x \in \mathbb{R}^n \mid Bx = d,\ x \geq 0 \text{ integer}\}. \end{cases}$$

Lagrangian dual and LP relaxation

The Lagrangian dual is a *partially convexified* version of the ILP.

$$\text{conv}X \subseteq \{ x \in \mathbb{R}^n \mid Bx = d,\ x \geq 0\} = \bar{X}$$

Now consider the continuous relaxation of the ILP.

$$\begin{cases} z_{LP} = & \min & c^T x \\ \text{s.t.} & & Ax = b, \\ & & x \in \bar{X}, \end{cases}$$

it is

$$z_{LD} \geq z_{LP},$$

- The best lower bound provided by Lagrangian relaxation is not worse than the LP bound.
- If the vertices of $\bar{X}$ are integer, it is $z_{LD} = z_{LP}$ (*integrality property*).

Outline

1. Lagrangian relaxation and related topics
   - Tackling the Lagrangian dual (LD)

Tackling the Lagrangian dual

Previous formulations offer different solution schemes:

Nonsmooth formulation:

- Dual ascent.
- Subgradient and variants.
- Cutting plane.

LP formulation:

- Column generation.

Dual ascent-Subgradient

Maximise $z_{LR}(\lambda)$ either:

- Changing at each iteration just one multiplier, to get an increase in the dual function.

or

- Adopting the subgradient method (typically equipped with Polyak step). Note that, once for any λ the solution $x(\lambda)$ has been obtained, it is

$$g(\lambda) = b - Ax(\lambda) \in \partial z_{LR}(\lambda).$$

Cutting plane

- Take any subset of points $x_k \in X$, $k \in \mathcal{S}$ (the *bundle*), generate an (upper) approximation of the min function and solve the *restricted primal* (assumed to be not unbounded!))

$$z_{restr}(\mathcal{S}) = \max_{\lambda} \min_{k \in \mathcal{S}} c^T x_k + \lambda^T (b - A x_k).$$

- Let $\lambda_{\mathcal{S}}$ be any optimal solution. Solve the Lagrangian relaxation for $\lambda = \lambda_{\mathcal{S}}$.
- Let $x_{k_{\mathcal{S}}}$ be the optimal solution of the relaxation and assume $z_{LR}(\lambda_{\mathcal{S}})$ be sufficiently smaller than $z_{restr}(\mathcal{S})$, then augment the bundle by $x_{k_{\mathcal{S}}}$ and iterate.

Column generation

Take again the LP formulation of the Lagrangian dual (the variables are the μ_k, $k = 1, \ldots, K$).

$$\begin{cases} z_{LD} = \min\limits_{\mu} \quad c^T(\sum\limits_{k=1}^{K} \mu_k x_k) \\ \text{s.t.} \qquad A(\sum\limits_{k=1}^{K} \mu_k x_k) = b \\ \qquad\qquad \sum\limits_{k=1}^{K} \mu_k = 1, \quad \mu_k \geq 0, \ k = 1, \ldots, K, \end{cases}$$

A (possibly very large) Linear Program suitable for column generation.
The $(m+1) \times K$ constraint matrix is :

$$\begin{bmatrix} Ax_1, & \ldots, & Ax_K \\ 1 & \ldots & 1. \end{bmatrix}$$

Column generation

- Columns associated to points x_ks, in perfect analogy with the association affine pieces–points of X in the nonsmooth formulation.
- Start from any subset of columns providing a basic feasible solution and then look for a column with negative reduced cost.
- Solve a *pricing* problem. Letting λ_B be the vector of the first m dual variables, the components of the reduced cost vector $\hat{c}$ are:

$$\hat{c}_k = (c_k - A^T \lambda_B)^T x_k - \lambda_{m+1}, \quad k = 1, \ldots, K,$$

λ_{m+1} is the last dual variable and is a constant in the definition of $\hat{c}_k$. The pricing problem consists of solving

$$\min_{k=1,\ldots,K} (c_k - A^T \lambda_B)^T x_k,$$

which is exactly the Lagrangian relaxation for $\lambda = \lambda_B$.

Some remarks

- If the integrality property holds, Lagrangian relaxation provides a poor bound. Nevertheless
  - The LP relaxation might be huge.
  - In Lagrangian relaxation the decision variables keep the original physical meaning.
  - Possibly infeasible solutions of the Lagrangian relaxation are often more suitable for *repairing* heuristics.
- In case complicating constraints are of type $Ax \geq b$, multiplier vector λ must be nonnegative, thus LD is a constrained problem. If $x(\lambda)$ is feasible, it is only ϵ-optimal, for

$$\epsilon = (Ax(\lambda) - b)^T \lambda \geq 0.$$

- It is not strictly necessary to solve exactly the Lagrangian relaxation!

Outline

1. Lagrangian relaxation and related topics
   - Lagrangian heuristics

Basic scheme

Lagrangian heuristic

Step 0 (*Initialization*) Choose $\lambda^{(0)}$. Set set $k = 0$ and $z_{UB} = \infty$.

Step 1 (*Lagrangian relaxation*) Calculate $z_{LR}(\lambda^{(k)})$ and the correspondent $x(\lambda^{(k)})$. If $x(\lambda^{(k)})$ is feasible then STOP.

Step 2 (*Repairing heuristic*) Implement any heuristic algorithm to provide, starting from $x(\lambda^{(k)})$, a feasible solution $x_{eur}^{(k)}$. Set $z_{eur}^{(k)} = c^T x_{eur}^{(k)}$ and possibly update z_{UB}.

Step 3 (*Multiplier update*) Calculate $\lambda^{(k+1)})$ by applying any algorithm for the Lagrangian dual. Set $k = k + 1$. Perform a termination test and possibly return to Step 1.

Outline

1. Lagrangian relaxation and related topics
   - An example: The Set Covering problem (SC)

A dual ascent heuristic for SC

Given:

- A ground-set $I = \{1, \ldots, m\}$
- A family of n subsets $\mathcal{S}_j \subseteq I$, $j \in J = \{1, \ldots n\}$
- A cost vector $c \in \mathbb{R}^n$, $c > 0$

Find a minimum cost cover, that is an index set $J^* \subseteq J$ such that $\cup_{j \in J^*} \mathcal{S}_j = I$ with minimal associated cost $\sum_{j \in J^*} c_j$. Letting A be the $m \times n$ binary incidence matrix of the subsets $\mathcal{S}_j$ and e a vector of m ones, the SCP reads as follows

$$\begin{cases} z_{SC} = & \min \quad c^T x \\ & \text{s.t.} \quad Ax \geq e, \\ & x \text{ binary}, \end{cases}$$

Relax the covering constraints:

$$\begin{cases} z_{LR}(\lambda) = & \min \; c^T x + \lambda^T (e - Ax) \\ & \text{s.t.} \quad x \text{ binary}. \end{cases}$$

Rewrite the Lagrangian relaxation as:

$$\begin{cases} z_{LR}(\lambda) = e^T\lambda + & \min(c - A^T\lambda)^T x \\ & \text{s.t.} \quad x \text{ binary.} \end{cases}$$

The solution is obtained by simple inspection of the sign of the reduced costs:

$$x_j(\lambda) = \begin{cases} 1 & \text{if } c_j(\lambda) = c_j - a_j^T\lambda \leq 0 \\ 0 & \text{otherwise,} \end{cases}$$

where a_j is the jth column of A.
Now, in case $x(\lambda)$ is infeasible, we introduce a rule for updating just one component of λ so that:

- the number of satisfied constraints is increased.
- the function z_{LR} increases as well.

$x(\lambda)$ infeasible $\Rightarrow$ there exists at least one row index, say h, such that:

$$\sum_{j \in J} a_{hj} x_j(\lambda) = 0,$$

which implies, in turn,

$$c_j(\lambda) = c_j - a_j^T\lambda > 0, \quad \text{for all } \; j \in J^{(h)} = \{j \mid a_{hj} = 1\}.$$

Defining the new multiplier setting as $\lambda^+ = \lambda + \delta e_h$ for some $\delta > 0$, where e_h is the h-th unit vector, the updated reduced cost is $c(\lambda^+) = c(\lambda) - \delta A^T e_h$, that is

$$c_j(\lambda^+) = \begin{cases} c_j(\lambda) - \delta, & \text{if } j \in J^{(h)}, \\ c_j(\lambda), & \text{otherwise.} \end{cases}$$

In particular, by setting $\delta = \min_{j \in J^{(h)}} c_j(\lambda) = c_{j^*}(\lambda)$, it is $c_{j^*}(\lambda^+) = 0$ and thus$x_{j^*}(\lambda^+) = 1$. Summing up it is

$$x_j(\lambda^+) = \begin{cases} x_j(\lambda), & \text{if } j \neq j^*, \\ 1, & \text{if } j = j^*. \end{cases}$$

Corresponding to the new solution $\vec{x}(\lambda^+)$, the constraint h is no longer violated and it is $z_{LR}(\lambda^+) = z_{LR}(\lambda) + \delta$. Thus we have obtained:

- dualascent;
- at least one of the constraints previously violated is satisfied;
- the satisfied ones remain such.

In conclusions a feasible solution and an improving lower bound are obtained in at most m ascent steps.

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual
   - Tackling the Lagrangian dual (LD)
   - Lagrangian heuristics
   - An example: The Set Covering problem (SC)
2. Applications
   - Generalized assignment
   - Spanning tree with minimum branch vertices (ST-MBV)
   - Berth allocation problem in maritime terminals (BAP)
   - Cross docking
3. Conclusions
4. References

Outline

2 Applications
- Generalized assignment

Generalized assignment (GAP)

To assign jobs to machines with limited amount of a resource (e.g. time or space), while maximizing the value of the assignment.
I and J: set of machine and job indices, respectively.

- a_{ij}, the resource required by job j when processed on machine i, $i \in I$ and $j \in J$;
- b_i, resource availability of machine i, $i \in I$;
- c_{ij}, value of assigning job j to machine i, $i \in I$ and $j \in J$.

Variables: $x_{ij} = 1$ if job j is assigned to machine i, $x_{ij} = 0$ otherwise.

$$\begin{cases} \text{maximize} & \displaystyle\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ \text{subject to} & \displaystyle\sum_{i \in I} x_{ij} = 1, \quad j \in J, \\ & \displaystyle\sum_{j \in J} a_{ij} x_{ij} \leq b_i, \quad i \in I, \\ & x_{ij} \text{ binary}, \quad i \in I, \ j \in J. \end{cases}$$

GAP Decomposition

Relaxing the semi-assignment constraints $\sum_{i \in I} x_{ij} = 1$, $j \in J$, by the multiplier vector λ, define the upper bound $z_{LR}(\lambda)$

$$z_{LR}(\lambda) = \sum_{j \in J} \lambda_j + \text{ max } \sum_{i \in I} \sum_{j \in J} (c_{ij} - \lambda_j) x_{ij}$$

$$\text{subject to } \sum_{j \in J} a_{ij} x_{ij} \leq b_i, \quad i \in I,$$

$$x_{ij} \text{ binary}, \quad i \in I, \ j \in J.$$

The problem decomposes into $|I|$ binary knapsack subproblems, that is

$$z_{LR}(\lambda) = \sum_{j \in J} \lambda_j + \sum_{i \in I} z_{LR}^{(i)}(\lambda),$$

where

$$z_{LR}^{(i)}(\lambda) = \text{ max } \sum_{j \in J} (c_{ij} - \lambda_j) x_{ij}$$

$$\text{subject to } \sum_{j \in J} a_{ij} x_{ij} \leq b_i,$$

Multiplier update

- Setting the initial values of $\lambda_j = \max^{(2)}_{i \in I} c_{ij}$ ($\max^{(2)}$ indicates the second largest number in a set), guarantees that only infeasibilities of the type $\sum_{i \in \mathcal{I}} x_{ij}(\lambda) = 0$ can occur.
- The multiplier update is driven by such infeasibility. Consider any (unassigned) job h such that $\sum_{i \in I} x_{ih}(\lambda) = 0$ and observe that decrease in multiplier λ_h makes the reduced costs $(c_{ih} - \lambda_h)$ increase, $i \in I \Rightarrow$ job h becomes more competitive in view of possible assignment.
- It is possible to calculate exactly the minimum reduction Δ_h of λ_h which allows assignment of job h to at least one machine.

Calculation of Δ_h

Calculate Δ_{ih}, theminimum reduction in λ_h to allow assignment of job h to machine i, $i \in I$. Solve first

$$z_{LR}^{(i,h)}(\boldsymbol{\lambda}) = \max \sum_{j \in J,\, j \neq h} (c_{ij} - \lambda_j) x_{ij}$$

$$\text{subject to} \sum_{j \in J,\, j \neq h} a_{ij} x_{ij} \leq b_i - a_{ih},$$

$$x_{ij} \text{ binary}, \quad j \in J,\, j \neq h,$$

that is the optimal value on the ith knapsack on the remaining jobs if it is imposed the assignment of job h:

$$\Delta_{ih} = z_{LR}^{(i)}(\lambda) - \underbrace{\left(c_{ih} - \lambda_h + z_{LR}^{(i,h)}(\lambda)\right)}_{\text{optimal value if } x_{ih} = 1} \geq 0.$$

Finally set $\Delta_h = \min_{i \in I} \Delta_{ih}$. In case $\Delta_h > 0$, letting $\lambda^+ = \lambda - \Delta_h e_h$, function z_{LR} reduces of exactly Δ_h.

The Lagrangian dual

- At each iteration just one multiplier is updated (thus the algorithm is a co-ordinate descent one).
- Unlike the set covering algorithm, it is not ensured that the number of satisfied constraints increases monotonically.
- Termination at a feasible (and hence optimal) solution is guaranteed.

Outline

2 Applications
- Spanning tree with minimum branch vertices (ST-MBV)

The problem

Design of optical networks.

- Construct a spanning tree to guarantee connection to all vertices of a given network
- At each vertex of the tree with more than two incidents edges (degree > 2) a device(switch) is to be installed.

Thus

- Find a spanning tree while minimizing the number of branch vertices.

Given the undirected network $G = (V, E)$ (V set of n vertices and E set of m edges) define the decision variables:

- x_e, $e \in E$, binary; $x_e = 1$ if edge e is selected and $x_e = 0$ otherwise;
- y_v, $v \in V$, binary; $y_v = 1$ if v is a branch vertex; $y_v = 0$ otherwise.

The model

The formulation:

$$\begin{cases} \text{minimize} & \displaystyle\sum_{v \in V} y_v \\ \text{subject to} & \displaystyle\sum_{e \in E} x_e = n - 1, \\ & \displaystyle\sum_{e \in E(S)} x_e \leq |S| - 1, \ S \subseteq V, \\ & \displaystyle\sum_{e \in A(v)} x_e - 2 \leq \delta_v y_v, \ v \in V, \\ & y_v \text{ binary}, \ v \in V \text{ and } x_e \text{ binary}, \ e \in E, \end{cases}$$

- For any given subset of vertices S, $E(S)$ is the set of edges with both the endpoints in S. $A(v)$ denotes the set of incident edges to vertex v, with $\delta_v = |A(v)|$.
- The objective function is the total number of branch vertices.
- Note that $\displaystyle\sum_{e \in A(v)} x_e > 2 \ \Rightarrow \ y_v = 1$

The Lagrangian relaxation

Introducing the multiplier vector λ, obtain the Lagrangian relaxation:

$$z_{LR}(\boldsymbol{\lambda}) = \text{min} \quad \sum_{v \in V} y_v + \sum_{v \in V} \lambda_v \Big(\sum_{e \in A(v)} x_e - 2 - \delta_v y_v \Big)$$

$$\text{subject to} \quad \sum_{e \in E} x_e = n - 1,$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \; S \subseteq V,$$

$$y_v \text{ binary}, \quad v \in V \text{ and } x_e \text{ binary}, \quad e \in E,$$

Problem above decomposes into two problems, in the variables y_v and x_e, respectively:

$$z_{LR}(\lambda) = -2 \sum_{v \in V} \lambda_v + z_{LR}^{(1)}(\lambda) + z_{LR}^{(2)}(\lambda),$$

where $z_{LR}^{(1)}(\lambda)$ and $z_{LR}^{(2)}(\lambda)$ are defined as follows:

$$z_{LR}^{(1)}(\lambda) = \min_y \text{binary} \quad \sum_{v \in V} y_v(1 - \delta_v \lambda_v)$$

and

$$z_{LR}^{(2)}(\lambda) = \min_y \text{binary} \quad \sum_{v \in V} \sum_{e \in A(v)} \lambda_v x_e$$

$$\text{subject to} \quad \sum_{e \in E} x_e = n - 1,$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \ S \subseteq V.$$

Note that both $z_{LR}^{(1)}(\lambda)$ and $z_{LR}^{(2)}(\lambda)$ are easy to calculate. In fact $z_{LR}^{(1)}(\lambda)$ is obtained by cost coefficients inspection, by setting

$$y_v = 1, \text{ if } \ 1 - \delta_v \lambda_v \leq 0, \ y_v = 0, \text{otherwise},$$

while $z_{LR}^{(2)}(\lambda)$ is the optimal value of the standard minimum spanning tree problem where the weight of edge $e = (u, v)$ is $\lambda_u + \lambda_v$.

The Lagrangian dual (sketch)

It is possible to devise an ascent strategy modifying one multiplier at a time. Consider the two following cases at $(x(\lambda), y(\lambda))$ for some $v \in V$:

1. $\sum_{e \in A(v)} x_e(\lambda) > 2$ and $y_v(\lambda) = 0$. In this case $\sum_{e \in A(v)} x_e(\boldsymbol{\lambda}) - 2 - \delta_v y_v(\lambda)) > 0$ is the v component of a subgradient of z_{LR} and it is strictly positive, thus an increase in the objective is expected if λ_v is increased.
2. $\sum_{e \in A(v)} x_e(\lambda) \leq 2$ and $y_v(\lambda) = 1$. In this case the vth subgradient component $\sum_{e \in A(v)} x_e(\lambda) - 2 - \delta_v y_v(\lambda)$ is negative and, in addition it is $\lambda_v > 0$. Thus a decrease λ_v is in order.

- The co-ordinate search approach is definitely faster than standard subgradient but the lower bound is slightly worse due to possible premature stop.
- A feasible solution of ST-MBV is available with no additional computational cost at each iteration of the dual ascent procedure.

Outline

2 Applications
- Berth allocation problem in maritime terminals (BAP)

The problem

Decide the allocation in space and time for all incoming ships in a given planning time horizon.

- To assign to incoming ships the berthing points along the quay.
- To provide for each berth the processing order of the assigned ships.
- The handling time of each ship depends on the berth.
- Ships arrive at different times.
- Berths are not necessarily available at the beginning of the planning time horizon.

To minimize the total turnaround time.

Formulation

Scheduling problem:

- Unrelated parallel machines (berths).
- Job release times (Ship arrival times).
- Machine availability constraints.

Problem data and notation

- N: set of ships to be berthed; $|N| = n$;
- M: set of berthing points; $|M| = m$;
- P: ordered set of positions; $|P| = |N| = n$;
- s^k: starting time for berth k;
- a_i: arrival time of ship i;
- t_i^k: handling time of ship i at berth k;
- $N(k) = \left\{ i \in N \mid a_i > s^k \right\} \ \forall k \in M$: set of ships arriving after starting time of berth k;
- $P(p) = \{ q \in P \mid q < p \} \ \forall p \in P$: set of positions preceding position p.

Decision variables

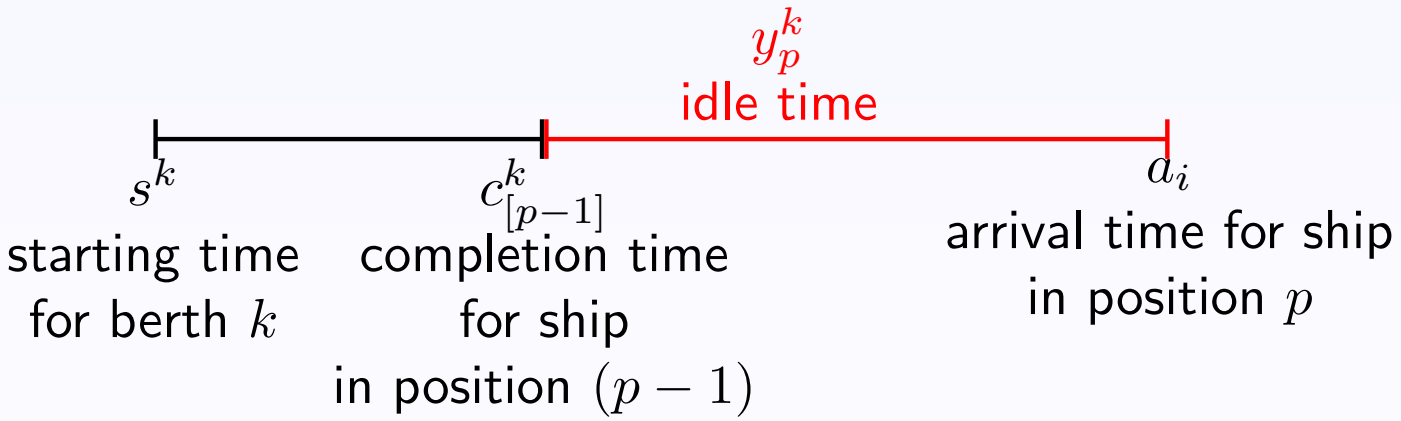
The scheduling variables are:

$$x_{ip}^{k} = \begin{cases} 1 & \text{if ship } i \text{ is the } p\text{-th ship moored at berth } k \\ 0 & \text{otherwise} \end{cases}$$

The following variables take into account the interplay between ship arrival times and berth availability:

y_p^k : idle time at berth k between ships $(p-1)$ and p.

Idle time


y_p^k
idle time
s^k
$c^k_{[p-1]}$
d_i
starting time for berth k
completion time for ship in position $(p-1)$
arrival time for ship in position p

The model and the complicating constraints

$$
\begin{cases}
\min \displaystyle\sum_{k\in M}\sum_{i\in N}\sum_{p\in P}\left\{(n-p+1)t_{ip}^k + s^k - a_i\right\}x_{ip}^k + \sum_{k\in M}\sum_{p\in P}(n-p+1)y_p^k \\
\text{s.t.} \\
\displaystyle\sum_{k\in M}\sum_{p\in P} x_{ip}^k = 1, \quad i \in N \\
\displaystyle\sum_{i\in N} x_{ip}^k \le 1, \quad k \in M,\ p \in P \\
\color{red}{y_p^k \ge \displaystyle\sum_{i\in N(k)}(a_i - s^k)x_{ip}^k - \sum_{l\in P(p)}\left(y_l^k + \sum_{j\in N} t_j^k x_{jl}^k\right), \quad k \in M,\ p \in P} \\
x_{ip}^k \in \{0,1\}, \quad k \in M,\ i \in N,\ p \in P; \quad y_p^k \ge 0, \quad k \in M,\ p \in P
\end{cases}
$$

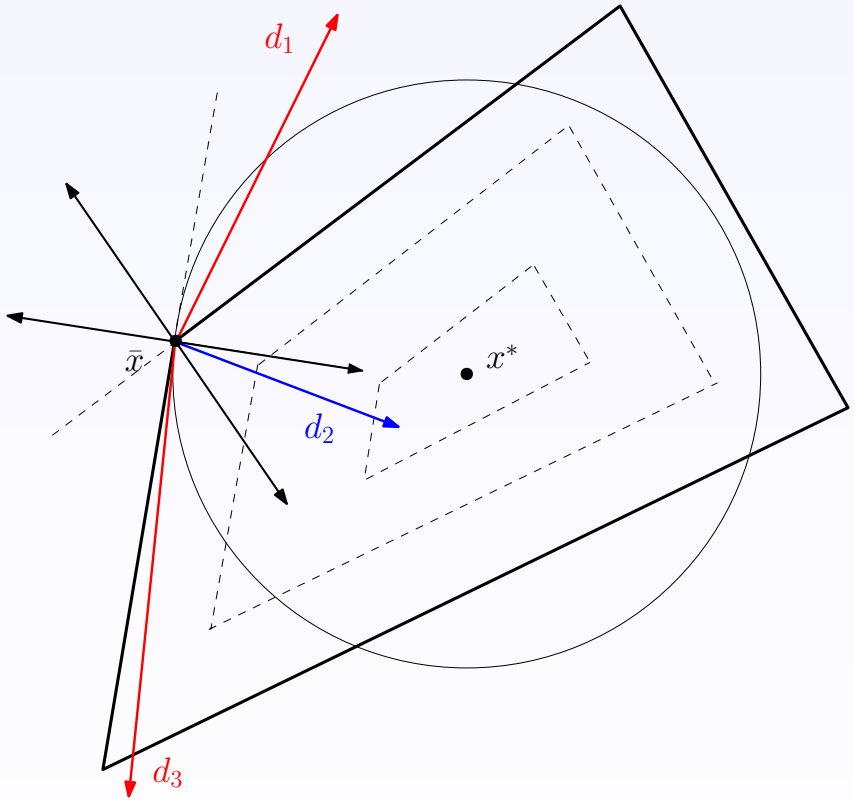
Remark

$N(k) = \emptyset,\ \forall k \Rightarrow$ *No idle times (static BAP: an assignment problem)*

The Lagrangian relaxation and the Lagrangian heuristic

- Judicious choice of the multipliers λ_{kp} to avoid the trivial bound $-\infty$.
- Constraints of the relaxed problem are the same as in the static BAP.
- Dual ascent (single multiplier update) equipped by an Armijo-type line search (with possible null–step!).
- Easy repairing of the solution of the relaxation (just insert idle times whenever needed).
- Excellent numerical results due to stronger formulation of BAP w.r.t. previous formulations.

Null step



Outline

2 Applications
- Cross docking

Scheduling at cross docking gates

- A cross docking (CD) centre is a logistic node where goods are unloaded from inbound trucks, consolidated and immediately loaded on outbound trucks.
- Traditional and costly storing and retrieval phases cancelled.
- Truck arrival and departure processes are strongly coupled.
- Synchronized scheduling of the inbound and outbound trucks.

The problem

- Find the optimal inbound and outbound truck sequences.
- Only two gates, one for the inbound and the other for the outbound trucks.
- The objective is to minimize the total completion time (the *makespan*).

Rules of the game

- Only one truck at a time can be handled at each gate; no *preemption* is allowed;
- Loading of an outbound truck cannot start until all goods it is expected to deliver have been unloaded;
- All trucks require the same processing time (one *slot*) and are ready at the time 0.

Data

- $I = \{1, \ldots, n\}$ and $J = \{1, \ldots, m\}$: sets of inbound and outbound trucks, respectively.
- J_i: the set of outbound trucks receiving goods from the inbound truck i, $i \in I$.
- I_j: the set of inbound trucks providing goods to the outbound truck j, $j \in J$.
- $K = \{1, \ldots, n\}$ and $L = \{1, \ldots, H\}$, $H \geq m + n$, be the time horizon for the inbound and outbound services, respectively.

Decision variables

- $x_{ik} = 1$ if the inbound truck i is assigned to the time-slot k and $x_{ik} = 0$ otherwise, $i \in I$, $k \in K$.
- $y_{jh} = 1$ if the outbound truck j is assigned to the time-slot h, $y_{jh} = 0$ otherwise, $j \in J$, $h \in L$.
- C_{Max}, the makespan

Formulation

$$
\begin{cases}
z = & \min C_{Max} \\
& \text{subject to } \sum_{k \in K} x_{ik} = 1, \quad i \in I \\
& \qquad \sum_{i \in I} x_{ik} = 1, \quad k \in K \\
& \qquad \sum_{h \in L} y_{jh} = 1, \quad j \in J \\
& \qquad \sum_{j \in J} y_{jh} \leq 1, \quad h \in L \\
& \qquad C_{Max} \geq \sum_{h=1}^{H} h y_{jh}, \quad j \in J \\
& \qquad \sum_{h=1}^{H} h y_{jh} \geq \sum_{k=1}^{n} k x_{ik} + 1, \quad j \in J,\ i \in I_j \\
& x_{ik} \text{ binary}, \ i \in I, \ k \in K; \quad y_{jh} \text{ binary}, \ j \in J, \ h \in L
\end{cases}
$$

Lagrangian relaxation

$$
\begin{cases}
z_{LR}(\lambda) = \min \left\{ C_{Max} + \sum_{j \in J} \sum_{i \in I_j} \lambda_{ij} \left(\sum_{k \in K} k x_{ik} - \sum_{h \in L} h y_{jh} + 1 \right) \right\} \\
\text{subject to } \sum_{k \in K} x_{ik} = 1, \quad i \in I \\
\qquad \sum_{i \in I} x_{ik} = 1, \quad k \in K \\
\qquad \sum_{h \in L} y_{jh} = 1, \quad j \in J \\
\qquad \sum_{j \in J} y_{jh} \leq 1, \quad h \in L \\
\qquad C_{Max} \geq \sum_{h=1}^{H} h y_{jh}, \quad j \in J \\
x_{ik} \text{ binary}, \; i \in I, \; k \in K; \quad y_{jh} \text{ binary}, \; j \in J, \; h \in L
\end{cases}
$$

Decomposition (1)

From

$$i \in I_j \Leftrightarrow j \in J_i,$$

setting:

$$s = \sum_{j \in J} \sum_{i \in I_j} \lambda_{ij} = \sum_{i \in I} \sum_{j \in J_i} \lambda_{ij},$$

$$\rho_i = \sum_{j \in J_i} \lambda_{ij}, \ i \in I; \qquad \sigma_j = \sum_{i \in I_j} \lambda_{ij}, \ j \in J,$$

z_{LR} becomes:

$$\begin{aligned} z_{LR}(\rho, \sigma) =& s + \min \Big\{ \sum_{i \in I} \sum_{k \in K} k \rho_i x_{ik} + C_{Max} - \sum_{j \in J} \sum_{h \in L} h \sigma_j y_{jh} \Big\} = \\ =& s + z_{LR}^{(1)}(\rho) + z_{LR}^{(2)}(\sigma). \end{aligned}$$

Decomposition (2)

$$\begin{cases} z_{LR}^{(1)}(\rho) = & \min \sum_{i \in I} \sum_{k \in K} k \rho_i x_{ik} \\ & \text{subject to } \sum_{k \in K} x_{ik} = 1, \quad i \in I \\ & \qquad \sum_{i \in I} x_{ik} = 1, \quad k \in K \\ & x_{ik} \text{ binary}, \ i \in I, \ k \in K, \end{cases}$$

$$\begin{cases} z_{LR}^{(2)}(\sigma) = & \min C_{Max} - \sum_{j \in J} \sum_{h \in L} h \sigma_j y_{jh} \\ & \text{subject to } \sum_{h \in L} y_{jh} = 1, \quad j \in J \\ & \qquad \sum_{j \in J} y_{jh} \leq 1, \quad h \in L \\ & \qquad C_{Max} \geq \sum_{h=1}^{H} h y_{jh}, \quad j \in J \\ & y_{jh} \text{ binary}, \ j \in J, \ h \in L. \end{cases}$$

Solving the relaxation

- Two simple single machine scheduling problems.
- Solutions obtainable by sorting the vectors ρ and σ, respectively
- There exists an optimal solution to the LD such that

$$s = \sum_{i \in I} \sum_{j \in J_i} \lambda_{ij} = \sum_{i \in I} \rho_i = \sum_{j \in J} \sum_{i \in I_j} \lambda_{ij} = \sum_{j \in J} \sigma_j \leq 1. \qquad (1)$$

- Optimal multiplier property helps in choosing the stepsize in the line search.
- Repairing heuristic based on forward-shifting of those outbounds trucks j for which the relaxed constraint

$$\sum_{h=1}^{H} h y_{jh} \geq \sum_{k=1}^{n} k x_{ik} + 1$$

has been violated for at least one $i \in I$.

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual
   - Tackling the Lagrangian dual (LD)
   - Lagrangian heuristics
   - An example: The Set Covering problem (SC)
2. Applications
   - Generalized assignment
   - Spanning tree with minimum branch vertices (ST-MBV)
   - Berth allocation problem in maritime terminals (BAP)
   - Cross docking
3. Conclusions
4. References

Conclusions

- Lagrangian relaxation + Effective ascent + Heuristics is a powerful combination of techniques.
- Promising opportunities in several other application fields.
- Invest on ascent methods based on inexact function evaluation.
- Extend to MINLP.

Outline

1. Lagrangian relaxation and related topics
   - The Lagrangian dual
   - Tackling the Lagrangian dual (LD)
   - Lagrangian heuristics
   - An example: The Set Covering problem (SC)
2. Applications
   - Generalized assignment
   - Spanning tree with minimum branch vertices (ST-MBV)
   - Berth allocation problem in maritime terminals (BAP)
   - Cross docking
3. Conclusions
4. References

Reference

M. Gaudioso, *A view of Lagrangian relaxation and its applications*, in Numerical nonsmooth optimization. State of the art algorithms, A.M. Bagirov, M. Gaudioso, N. Karmitsa, M. Mäkelä and S. Taheri, eds., Springer 2020, pp. 579–617.