Level bundle method applied to Quadratic Min-Cost Flow problem

Simone Baccile, `s.baccile@studenti.unipi.it`
Matteo D'Onofrio, `m.donofrio7@studenti.unipi.it`
Project NoML-15 of C.M. course 2021/22, University of Pisa

October 3, 2022

1 Introduction

Let the convex quadratic separable Min-Cost Flow (MCF) Problem

$$\min_x \quad x^T Q x + qx \tag{1a}$$

$$\text{s.t.} \quad Ex = b, \tag{1b}$$

$$0 \le x \le u \tag{1c}$$

where $Q \in \mathbb{R}^{n \times n}$ is diagonal and positive semidefinite ($Q \succeq 0$, so $diag(Q) \ge 0$), and $E \in \mathbb{R}^{m \times n}$ is the node-arc directed connected graph $G = (N, A)$

$$E_{i,j} = \begin{cases} -1 & \text{if arc } j \text{ leaves node } i \\ 1 & \text{if arc } j \text{ enters node } i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We want to solve the problem 1 using a dual approach, where the Lagrangian Dual is solved by a level bundle method algorithm.

1.1 Preliminary analysis

The matrix Q is diagonal and positive semidefinite, hence the function 1a is convex because Q is exactly its Hessian matrix. Q is also symmetric and all of its eigenvalues $\in \mathbb{R}$. Because Q is positive semidefinite can happen that it has eigenvalues $\lambda_i = 0$ which implies Q singular. The matrix $E \in \mathbb{R}^{m\times n}$ and it is a large sparse matrix with a lot of zero values. The graph G has m nodes and n arcs, so in general $n > m$, but it can also be $n \gg m$.

2 Dual Approach to Quadratic MCF Problem

In this section, we will describe how to solve the quadratic MCF problem 1 using a dual approach. First, in the subsection 2.1, we will apply a partial Lagrangian relaxation to simplify the problem, then in the subsection 2.2 we will solve the problem using a level bundle method.

2.1 Partial Lagrangian relaxation

In this section, we will analyze different partial Lagrangian relaxations in order to simplify the MCF problem 1. The idea behind the partial Lagrangian relaxation is to solve one or more constraints "by hand" when these constraints are simple to solve due to the structure of the problem. In the subsection 2.1.1, we analyze the relaxation of both constraints of the problem 1b and 1c, in 2.1.2 we analyze the relaxation of box constraints 1c, and in 2.1.3 we analyze the relaxation of the equality constraints 1b.

2.1.1 Relaxation of both constraints

Relaxing both constraints, equality 1b, and box 1c, of the problem 1 we obtain the Lagrangian function

$$L(x, \mu, \lambda_1, \lambda_2) = x^T Qx + qx + \mu(Ex - b) - \lambda_1(u - x) - \lambda_2(x) \tag{3}$$

where the dual variable $\mu \in \mathbb{R}^m$ is related to equality constraints, and the dual variables $\lambda_1, \lambda_2 \in \mathbb{R}^n_+$ are related to box constraints. The Lagrangian function 3 has the same structure as the initial quadratic function 1a with the addition of three linear terms, and it is still a convex function. Due to its structure, to find a feasible x, we can simply compute the gradient of $L(x, \mu, \lambda_1, \lambda_2)$ and solve $\nabla L(x, \mu, \lambda_1, \lambda_2) = 0$.

$$\begin{aligned} \nabla L(x, \mu, \lambda_1, \lambda_2) &= 0 \\ 2Qx + q + \mu E + \lambda_1 - \lambda_2 &= 0 \\ 2Qx &= -q - \mu E - \lambda_1 + \lambda_2 \\ x &= \frac{1}{2} Q^{-1}(-q - \mu E - \lambda_1 + \lambda_2) \end{aligned} \tag{4}$$

x can be found in quadratic time using this approach. After finding it, we can define the Dual Problem:

$$
\begin{aligned}
\max_{\mu, \lambda_1, \lambda_2} \quad & x^T Q x + qx + \mu(Ex - b) - \lambda_1(u - x) - \lambda_2(x) \\
\text{s.t.} \quad & \lambda_1, \lambda_2 \geq 0
\end{aligned}
\tag{5}
$$

which is a constrained maximization problem where the unknown variables (μ, λ_1, λ_2) are $m + 2n$.

2.1.2 Relaxation of box constraint

To relax the box constraints 1c of the problem 1, we split it into two inequalities: $(u - x) \geq 0$ and $x \geq 0$. Relaxing both inequalities, we obtain the Lagrangian function:

$$
L(x, \lambda_1, \lambda_2) = x^T Q x + qx - \lambda_1(u - x) - \lambda_2(x) \tag{6}
$$

where the dual variables λ_1 and $\lambda_2 \in \mathbb{R}^n_+$. To simplify the notation, when we talk about both variables, we will use $\lambda = \begin{bmatrix}\lambda_1 \ \lambda_2\end{bmatrix}$, where $\lambda \in \mathbb{R}^{2n}_+$. From the Lagrangian function we can derive the dual function respectively:

$$
\begin{aligned}
\psi(\lambda) = \ \min_{x} \quad & L(x, \lambda) \\
\text{s.t.} \quad & Ex = b
\end{aligned}
\tag{7}
$$

The Lagrangian function 6 has the same structure as the initial quadratic function 1a with the addition of two linear terms, and it is still a convex function. The derived dual function 7 has only equality constraints $Ex = b$ that we can solve explicitly using Poorman's KKT conditions that are sufficient for global optimality.

$$
\begin{cases}
\nabla L(x, \lambda) + \mu \nabla(Ex - b) = 0 \\
Ex = b
\end{cases}
\tag{8}
$$

$$
\begin{cases}
2Qx + E^T \mu = -q - \lambda_1 + \lambda_2 \\
Ex = b
\end{cases}
\tag{9}
$$

This system can be rewritten in the following matrix form

$$
\begin{bmatrix} 2Q & E^T \\ E & 0 \end{bmatrix}
\begin{bmatrix} x \\ \mu \end{bmatrix}
=
\begin{bmatrix} -q - \lambda_1 + \lambda_2 \\ b \end{bmatrix}
\tag{10}
$$

Solving the system, we find a feasible x that doesn't violate the constraints $Ex = b$, and μ which proves the optimality of x. The matrix $\tilde{Q} = \left[\begin{smallmatrix} 2Q & E^T \\ E & 0 \end{smallmatrix}\right]$ is $(n+m) \times (n+m)$, and both matrices $\left[\begin{smallmatrix} x \\ \mu \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} -q-\lambda_1+\lambda_2 \\ b \end{smallmatrix}\right]$ are $(n + m) \times 1$. To solve the system we can factorize the matrix $\tilde{Q}$ with LU factorization, which costs $\frac{2}{3}(n+m)^3$, and then solve it by back-substitution at the cost of $O\big((n+m)^2\big)$. The matrix $\tilde{Q}$ is factorized only once because Q and E do not change through iterations. While matrix $\left[\begin{smallmatrix} -q-\lambda_1+\lambda_2 \\ b \end{smallmatrix}\right]$ changes because at each iteration we try to have better λ. Indeed, the factorization cost $\frac{2}{3}(n+m)^3$ is paid once, while at each iteration the algorithm costs $O\big((n+m)^2\big)$ to find a feasible x.

After finding the feasible x, we can substitute it in the Lagrangian function 6 and define the Dual Problem:

$$\begin{aligned} \max_{\lambda_1, \lambda_2} \quad & x^T Q x + q x - \lambda_1 (u - x) - \lambda_2 (x) \\ \text{s.t.} \quad & \lambda_1, \lambda_2 \geq 0 \end{aligned} \tag{11}$$

which is a constrained maximization problem where the unknown variables (λ_1, λ_2) are $2n$.

2.1.3 Relaxation of equality constraints

Let's explore now the case in which we relax the equality constraint 1b. The resulting Lagrangian Function we obtain is the following:

$$L(x, \mu) = x^T Q x + q x + \mu (E x - b) \tag{12}$$

This is a function in two variables: $x \in \mathbb{R}^n$ and the Lagrangian multiplier $\mu \in \mathbb{R}^m$. In addition we can see that the Lagrangian function 12 has the same structure of the 1a with the addition of a linear term, so it's still a quadratic convex function. Hence the derived Dual function is:

$$\begin{aligned} \psi(\mu) = \min_{x} \quad & L(x, \mu) \\ \text{s.t.} \quad & 0 \leq x \leq u \end{aligned} \tag{13}$$

Now we need to find a solution x^* that is feasible concerning the box constraints, but before proceeding we make some considerations. The box constraints allow us to treat each variable x_i independently with respect to any other x_j with $i \neq j$, because the matrix Q is diagonal, so in our objective function we'll never have any linear combination between any pair of variables x_i, x_j. This assumption implies that 13 is a separable problem, that can be divided into n independent problems, so that each constraint can be solved separately from the others.
We know that the matrix Q is positive semidefinite, so based on the value of Q_{ii}, we can have two different cases:

- $\mathbf{Q_{ii} = 0}$: in this case, the function 12 is no longer quadratic, but a linear function. In fact, the function becomes $L(x, \mu) = qx + \mu(Ex - b)$ (to simplify the notation, we have omitted the index i for each variable). To find a feasible x which minimizes $L(x)$ into a closed interval $[0, u]$, after choosing a random μ, just evaluate the function in 0 and u and return as feasible $\bar{x}$ 0 if $L(0) < L(u)$, otherwise return u.

- $\mathbf{Q_{ii} > 0}$: in this case we proceed in two steps:

    1. Find an x^* solution by minimizing the Lagrangian function $L(x)$, without considering the box constraints where μ is randomly initialized (here too we omit the indices i of the single variables in order to simplify the notation). We can easily find the solution x^* by looking for the function's stationary point, so where $\nabla L(x) = 0$.

$$
\begin{aligned}
&\nabla L(x) = 0 \\
\iff\ & 2Qx + q + E^T\mu = 0 \\
\iff\ & x^* = \frac{1}{2}Q^{-1}(-q - E^T\mu)
\end{aligned}
\tag{14}
$$

2. Project x^* in the finite set X defined by the box constraints to get a feasible solution. To do this, we exploit the following theorem.

   Theorem 1. *Given the quadratic and convex function $f = x^TQx + qx + b$ (Q diagonal, $diag(Q) > 0$), the optimal solution x^* which minimizes f, and the finite set $X = [x_-, x_+]$, the optimal solution $\bar{x}$ which minimizes the function f restricted to X is given by* $\max(x_-, \min(x^*, x_+))$.

   Proof. To prove that $\bar{x} = \max(x_-, \min(x^*, x_+))$ we consider three different cases based on the value of x^* with respect to the interval X.

   - $\mathbf{x_- \leq x^* \leq x_+}$: this is a trivial case. $x^* \in X$, so f is already minimized inside the interval by its optimal solution. $\min(x^*, x_+) = x^*$, and so $\bar{x} = \max(x_-, x^*) = x^*$.
   - $\mathbf{x_- \leq x_+ < x^*}$: in this case, x^* is outside the interval, so we need to project x^* inside X to get a local solution that minimizes f. Since f is a convex function and in x^* it is minimized, we can say that from $-\infty$ to x^* f decreases, while from x^* to ∞ increases. The interval X is to the left of x^*, this means that f decreases. Since f decreases $f(x_+) \leq f(x_-)$, then the local minimum of f in X is x_+. Considering our hypothesis $x_- \leq x_+ < x^*$, $\min(x^*, x_+) = x_+$ and $\max(x_-, x_+) = x_+$. So $\bar{x} = x_+$.
   - $\mathbf{x^* < x_- \leq x_+}$: this case is like the previous one, x^* is outside the interval again, but this time X is to the right of x^* which means that the function f increases. Since f increases $f(x_-) \leq f(x_+)$, then the local minimum of f in X is x_-. Considering our hypothesis $x^* < x_- \leq x_+$, $\min(x^*, x_+) = x^*$ and $\max(x_-, x^*) = x_-$. So $\bar{x} = x_-$.

   □

   We can use this theorem to find a feasible solution $\bar{x}$ using the optimal solution x^*: $\forall i\ \bar{x}_i = \max(0, \min(x_i^*, u_i))$.

The expensive part of this method is finding the optimal x^* for $L(x)$ when $Q_{ii} > 0$. It takes $O(mn)$ computing $E^T\mu$, plus lower-order terms, to the worst case that is when Q is positive definite. The projection of x^* costs $O(n)$ because it consists of n comparisons. When $Q_{ii} = 0$ the feasible is found in constant time.
By substituting $\bar{x}$ in the Lagrangian function we can define the Dual Problem (D):

$$
\max_{\mu} \quad \bar{x}^TQ\bar{x} + q\bar{x} + \mu(E\bar{x} - b) \tag{15}
$$

which is an unconstrained maximization problem where we need to find the Lagrangian Multiplier $\mu \in \mathbb{R}^m$.

2.1.4 Relaxations comparison

After a cost and structure analysis of the problems obtained after relaxation, we decided to apply a Level Bundle method to the problem obtained from equality constraints relaxation described in 2.1.3. First of all because compared to other relaxations, the Dual Problem obtained is an unconstrained optimization problem (15), while the other two problems have their Lagrangian multipliers constrained in sign (5, 11). This makes the MP of the Bundle method easier. Then we also have a difference in the size of the dual variables. If we relax both constraints the Dual Problem has $m + 2n$ variables to find. Relaxing only the box constraints has $2n$ variables while relaxing only the equality constraints the variables are m. m is the number of nodes of the directed connected graph, while n is the number of arcs. In general, $n > m$, but can also be $n \gg m$.

2.2 Level Bundle method

In this section, we will describe the level bundle methods, a class of iterative algorithms used to solve the convex minimization problem. We will apply these algorithms to Dual Problem 15 to find an optimal solution and solve MCF Problem 1 using a dual approach. In the subsection 2.2.1, we will describe this category of algorithms to solve minimization problems, in 2.2.2 we will focus on a subclass of bundle methods called the level bundle method, in 2.2.3 we will introduce a dual approach of LBM to find a solution for the original problem, and in 2.2.4 we will analyze how to update the optimal variable through iterations and how to choose the level parameter.

2.2.1 Bundle methods

Bundle methods (BM) are a class of iterative algorithms that can be used to solve convex minimization problems. Since our Dual Problem 15 is a maximization problem, we can rewrite it in a minimization problem thanks to its structure in the following way:

$$\min_{\mu} \quad -x^T Qx - qx - \mu(Ex - b) \tag{16}$$

To simplify the notation we have rewritten $\bar{x}$ as x, maintaining all the considerations made in 2.1.3. Considering this problem a BM will produce a sequence μ^i of iterates which will possibly converge to an optimum μ^*. At each point μ^i the algorithm produces a pair $(f(\mu^i), z^i)$ where $f(\mu^i)$ is the function $-x^T Qx - qx - \mu(Ex - b)$ evaluated in μ^i, and $z^i \in \partial f(\mu^i)$ is the partial derivative. Because each pair is reused at each iteration, we want to avoid evaluating $f(\mu^i)$ every time because it can be costly. So instead of saving pairs $(f(\mu^i), z^i)$, we store pairs as (z^i, α^i) where $\alpha^i = \langle z^i, \mu^i\rangle - f(\mu^i)$. The pairs of each iteration are stored in the *Bundle* $\mathcal{B} = (z^i, \alpha^i)$. The *Bundle* is used to construct the *Cutting Plane Model* (CPM) of f:

$$f_{\mathcal{B}}(\mu) = \max\left\{\langle z^b, \mu\rangle - \alpha^b : b \in \mathcal{B}\right\} \tag{17}$$

The CPM $f_{\mathcal{B}}$ is a global lower model for f, in fact, $f_{\mathcal{B}} \leq f$. The CPM is used to define the *Master Problem* (MP) which can be solved to select the new iterate:

$$\mu^i \in \arg\min\{f^i_{\mathcal{B}}(\mu) : \mu \in M\} \tag{18}$$

In this case, the MP is a nonlinear optimization problem. To avoid solving a non-linear problem, we can rewrite 18 as an LP problem:

$$(\mu^i, v^i) \in \arg\min\{v : v \geq \langle z^b, \mu\rangle - \alpha^b, b \in \mathcal{B}^i, \mu \in M\} \tag{19}$$

The extra variable $v^i = f^i_{\mathcal{B}}(\mu^i)$. The MP 19 is easy to solve at least if $|\mathcal{B}^i|$ is small. The CPM is globally convergent. In fact, through iterates, the *Bundle* $\mathcal{B}$ grows, so that the CPM $f_{\mathcal{B}}$ becomes a better model until it converges to the real function f (because it is convex). The bad news is that while $|\mathcal{B}|$ grows, also the cost per iteration to solve the MP grows, and the convergence can be very slow.
The second issue arises from the search space. μ^i that we found at each iterate can be very far from μ^*, but also from two subsequent iterates we can have this problem. This is known also as the instability problem and it is another cause of the slowdown in convergence. Fortunately, the instability issue can be limited by introducing the stabilization for the MP. The objective of the stabilization is to produce iterates that improve the the value of objective function trying to avoid moving too far with respect to the previous best result. During the execution of a BM, the algorithm keeps a variable $\bar{\mu}$ called *stability center* which is the value of the best μ^i so far, so the one such that $f(\mu^i)$ has the lowest value found yet. In this reformulation, we show a simple approach called *Trust-region-stabilization* using the stability center. Here we ensure that each new iterate μ^i is near enough to $\bar{\mu}$, so is kept in the trust region around the current stability center. The stabilization is regulated by the stabilization parameter δ.

$$(\mu^i, v^i) \in \arg\min\{v : v \geq \langle z^b, \mu\rangle - \alpha^b, b \in \mathcal{B}^i, \|\mu - \bar{\mu}\| \leq \delta^i\} \tag{20}$$

2.2.2 Level Bundle method

The Level Bundle method (LBM) is a subclass of BMs which uses a different approach to select the new iterate in the MP. The idea behind this algorithm is the same as described previously, store at each iteration pairs (z^i, α^i) in $\mathcal{B}$, construct the CPM $f_{\mathcal{B}}(\mu)$ and solve the MP to get a new iterate (μ^i, v^i). In 20, the idea is to exclude from the search for the optimal solution the points where $f_{\mathcal{B}}(\mu) \ll f(\bar{\mu})$, but it underestimates the true value of f in the space. In this approach the amount of descent is regulated by the stabilization parameter δ. In LBM, the main idea is to use level stabilization fixing ahead of how much the model can descend. In this case, the descent is regulated by a level parameter $l < f(\bar{\mu})$ that restricts our options to choose a new iterate. But the set of points can be unbounded, so we also force the "closeness" to the stability center, obtaining the following MP:

$$\mu^i \in \arg\min\{\|\mu - \bar{\mu}^i\| : f^i_{\mathcal{B}}(\mu) \leq l^i\} \tag{21}$$

Let's now introduce a slightly different form based on the concept of *displacement*. The displacement can be defined as $d = \mu - \bar{\mu}$ and describes the gap between the stability center and the new iterate. The introduction of the displacement leads to modifying the pairs in the bundle. The pairs are always the same (z^i, α^i), but in this case, they are computed differently $z^i \in \partial f(\bar{\mu}+d)$, and $\alpha^i = \langle z^i, d\rangle - f(\bar{\mu}+d)$. The CPM is computed considering the displacement as $f_{\mathcal{B}}(\bar{\mu}+d)$. So the MP 21 can be rewritten taking into account d as follows:

$$d^i \in \arg\min \left\{ \|d\|_2^2/2 : f^i_{\mathcal{B}}(\bar{\mu}+d) \le l^i \right\} \tag{22}$$

As already seen in 19, this MP can be rewritten in the explicit form to exploit the LP property.

$$(d^i, v^i) \in \arg\min\{\|d\|_2^2/2 : v \ge \langle z^b, d\rangle - \alpha^b \;\; b \in \mathcal{B}^i, \;\; l^i \ge v\} \tag{23}$$

2.2.3 A dual approach to the Level Bundle Method

As seen in the paragraphs before, we've defined a way to minimize the MP by finding the pair (d^i, v^i). In this paragraph, we discuss a dual form of 23 that aims to find the variable d^i through the computation of a set of convex combinators to exploit all the information in the bundle. Convex combinators are non-negative coefficients whose sum is 1 that are used to compute a linear combination of vectors. We called those combinators θ. As far as we are concerned, θs are used to find at each iteration the dual variable μ, but under certain conditions (which we will explain later), they can also be used to retrieve a feasible primal solution for the problem 13.
First, we will see the dual form of problem 19, then, we'll describe the specific form of this problem for the MP 21, which uses the LBM. The dual problem of the MP 19 is the following:

$$\theta^i \in \arg\min\{\sum_{b\in B^i} \alpha^b\theta^b : \sum_{b\in B^i} z^b\theta^b = 0, \sum_{b\in B^i}\theta^b = 1, \theta^b \ge 0, b \in B^i\} \tag{24}$$

In this dual problem, there are several constraints that we are going to explain. First of all, immediately deduces that only the points present in the bundle are used to build the new iterate, and we do not have additional constraints that depend on the stability center (or at least not directly). The two constraints $\sum_{b\in B^i}\theta^b = 1$ and $\theta^b \ge 0$, as already mentioned, are used to define θ^i as a convex combinator. We also define the set of convex combinators Θ, which is the unitary simplex of appropriate dimension, and thanks to the constraints, we have immediately that each $\theta^i \in \Theta$. The last but not least constraints $\sum_{b\in B^i} z^b\theta^b = 0$ exploits the subgradients z^b using as much as possible accurate information about $\bar{\mu}$. So, if $z^b = 0$ we can already find a feasible solution, but if $z^b \neq 0$ we try to force the feasibility exploiting θ. Regarding the objective function $\sum_{b\in B^i}\alpha^b\theta^b$, it can be seen as a function that we are minimizing to obtain the smallest ϵ such that $0 \in \partial_\epsilon f^i_{\mathcal{B}}(\bar{\mu}) = \{z = \sum_{b\in B^i} z^b\theta^b = 0 : \theta \in \Theta, \sum_{b\in B^i}\alpha^b\theta^b \le \epsilon\}$.
Introducing the level parameter l of LBM, the dual problem 24 becomes slightly different. The dual of problem 21 is:

$$\theta^i \in \arg\min\{\|\sum_{b\in B} z^b\theta^b\|_2^2/2 + \sum_{b\in B}(l+\alpha^b)\theta^b : \theta \ge 0\} \tag{25}$$

This formulation has a few advantages. First of all, in the objective function, we are searching for a solution that has both a small norm $\|z^b\|$ and a small α^b. This is very different from the previous approach because in this case it is taken into consideration the "quality" of the first-order information w.r.t. $\bar{\mu}^i$, which previously was ignored. The level parameter l is also inserted into the objective function to exploit the local information of the problem preventing the model from going too far beyond the region with reasonable solutions. This parameter can also be used as a sort of "weight" by encouraging to only use z^b with "small" α^b (small l), or vice versa. The second advantage regards that dual problem 24 having the constraint $\sum_{b\in B^i} z^b\theta^b = 0$ could provide an empty solution, while this formulation always has a solution. Another useful consequence of the use of this dual problem is that we have an approximation of the subgradients of the function, that we can use as an "oracle" to see when the LBM has reached an optimal solution. In fact when $z^b = 0$ and $\alpha^b = 0$, $\bar{\mu}$ is the optimal solution.
A problem arising from this reformulation is that θ^i does not necessarily belong to Θ because we removed the constraint $\sum_{b\in B^i} \theta^b = 1$. But this is not a real problem because we can simply substitute θ^i with $\frac{\theta^i}{\langle \theta^i, e\rangle}$ which belongs in Θ (e is the vector of all ones). Another "issue" that we must take into account is that problem 25 could be unbounded below, as the objective function is not necessarily strictly convex. That also means that the MP 23 is empty. This issue can be controlled by updating the level parameter.
Once we find the dual variables θ^i, we can get back our displacement d^i as a linear combination of the weights θ and the subgradients z present in the bundle so far:

$$d^i = -\sum_{b\in B} z^b\theta^b \tag{26}$$

With d^i we can now compute the new $\mu^i = \bar{\mu} + d^i$, and its respective function value $v^i = f^i_B(\mu^i)$. Another advantage of problem 25 is that we can compute the new pair to be inserted into the bundle using θ. In fact, if $\theta^i \in \Theta$, we can insert the new aggregated pair $(z^i = z(\theta^i), \alpha^i = \alpha(\theta^i))$ in B, where $z(\theta^i) = \sum_{b\in B} z^b\theta^b$ and $\alpha(\theta^i) = \sum_{b\in B} \alpha^b\theta^b$.
Until now, what we have done is use θ to find a displacement and construct a new iterate with μ to find its relative x using the method described in 2.1.3. But what we can do is get back the primal solution directly using θ. We have three different ways to retrieve a primal solution for the problem:

1. If problem 25 produced θ^i such that $\theta^i \in \Theta$ and $\sum_i z^i\theta^i = 0$, then we would have that $\sum_i (Ex^i - b)\theta^i = 0$ by the definition of z. This summation can be rewritten as $E\sum x^i\theta^i = b\sum_i \theta^i$. As $\theta \in \Theta, \sum_i \theta^i = 1$, so $E\sum x^i\theta^i = b$. From here, we can derive our feasible solution because we are finding an x such that $Ex = b$, indeed $\bar{x} = \sum_i x^i\theta^i$ is a convex combination of all the primal solutions found so far. It's trivial to verify that $\bar{x}$ is feasible belonging to the range $[0, u]$. Knowing that each $x^i \in [0, u]$ because at each iteration, the algorithm produces a feasible solution, θ^i is always ≥ 0, so in any case, $\sum_i x^i\theta^i \geq 0$ for each i. In the case of the right boundary, to the worst case, we can have $\theta^i = 1$, but in any case, $x^i \leq u$, so for each i, $\sum_i x^i\theta^i \leq u$.

2. If $\sum_i z^i \theta^i \neq 0$, we can use a Lagrangian heuristic trying to force the two optimality conditions to get a feasible solution. First, we force θ to belong to Θ (if it doesn't belong to it) by rescaling θ so that $\sum_i \theta^i = 1$. Then we compute $\bar{x}$ as a convex combination of the previous x^i, and we try to force the second optimality condition, projecting it into subspace $Ex = b$. The projection can be done by solving:

$$x = \arg\min \Big\{ \frac{1}{2} \|x - \bar{x}\|_2^2 : Ex = b \Big\} \tag{27}$$

In order to solve this problem, we need to satisfy the "Poorman's KKT conditions" by finding x such that $Ex = b$ and finding λ such that $E^T\lambda = \nabla(\frac{1}{2}\|x - \bar{x}\|_2^2) \implies x - \bar{x} + E^T\lambda = 0$. That x can be found via closed formula using the pseudo-inverse of the matrix E:

$$\begin{aligned} & x - \bar{x} + E^T\lambda = 0 \\ \iff & EE^T\lambda = E\bar{x} - Ex \quad \text{(multiply by E both sides)} \\ \iff & EE^T\lambda = E\bar{x} - b \quad (Ex = b) \\ \iff & \lambda = (EE^T)^{-1}(E\bar{x} - b) \end{aligned} \tag{28}$$

$$x = \bar{x} - E^{+}(E\bar{x} - b) \quad \text{(substitute } \lambda) \tag{29}$$

where $E^+ = E^T(EE^T)^{-1}$.

At this point, we need to verify if this solution satisfies or not the box constraints. If $x \in [0, u]$, we have found our primal solution. If it doesn't happen, it leads us to the third case.

3. If we don't have any optimality condition satisfied and our Lagrangian heuristic failed to find a feasible primal solution, the only thing that we can do is to use θ to find the Lagrangian multiplier μ that we use to find a new iterate $\bar{x}^i$. This can be done using the same method described in section 2.1.3.

2.2.4 LBM steps

This is an iterative method we need rules to update the stability center $\bar{\mu}^i$ and the level parameter l^i. Regarding the update of the stability center, we proceed with the following approach. In each iterate, it finds a new solution, $\bar{\mu} + d$. If $f(\bar{\mu} + d) - f(\bar{\mu}) \leq m_1(v^i - f(\bar{\mu}))$, having $m_1 \in (0, 1)$, then an improvement is obtained, so we can do a *Serious Step* (SS) updating $\bar{\mu}^{i+1} = \bar{\mu}^i + d$; if instead, we've no improvement we do a *Null Step* (NS), leaving the stability center unchanged $\bar{\mu}^{i+1} = \bar{\mu}^i$.
For what concerns parameter l, there are many different ways for defining and updating it. The update of this parameter is called *Level Step* (LS).

- **Use of $\mathbf{f}_*$**: the best theoretical one is $l^i = \lambda f(\bar{\mu}^i) + (1 - \lambda)f_*$, with $\lambda \in (0, 1]$. Since we do not have the optimal value f_* available, we need to find the best possible approximation for this value.

- **Arbitrary l**: another possibility is to choose at the start l^i arbitrarily and update it if it leads to 21 empty. The update of the parameter is done until a non-empty region of the space is found, having $l^{i+1} > l^i$. Different rules can be used to update l^i. However, these rules take into account the distance between $f(\bar{\mu}^i)$ and l^i, then the update can be done weighting those values ($l^{i+1} = \lambda f(\bar{\mu}^i) + (1-\lambda)l^i$) or simply halving this gap ($l^{i+1} = l + \Delta/2$ where $\Delta = f(\bar{\mu}^i) - l^i$). We can also update l when the MP is not empty to reduce the search space. If we have the optimality condition satisfied and we can retrieve a primal solution, as written in cases 1 and 2 of paragraph 2.2.3, we can use that solution as an upper bound for the searching space. So, we simply set $l = f(\bar{x})$, because it is useless to look further for a solution above the upper bound.

- **Use of $\mathbf{v^i}$**: a better but more expensive approach is to choose at the start, as best lower bound for f the value v^i. Here the LS would be $l^{i+1} = \lambda f(\bar{\mu}^i) + (1-\lambda)v^i$. In this way, we have to solve two MPs: with the first unstabilized MP 19 we find v^i to update the level parameter, then we solve the second MP 25 to obtain a new iterate.

3 Algorithm

In this section, we will describe the algorithm for the LBM in 3.1 and its convergence analysis in 3.2.

3.1 Pseudocode

Algorithm 1 implements the process described in section 2.2. The algorithm is regulated by 4 parameters as below:

- *best_l*: a boolean parameter to decide whether to use the arbitrary level parameter (False) or the optimal value of the function obtained by solving the MP (True).

- l: the level parameter to use in the arbitrary case. It is ignored when *best_l* is true.

- λ: weight parameter used to update the level parameter. It is used only when the level parameter is computed considering v^i. $\lambda \in (0, 1]$.

- m: weight parameter that regulates the choice between NS and SS. $m \in (0, 1)$.

Algorithm 1 Level bundle method algorithm

Input f function to be minimized, $\bar{\mu}$ the starting point for μ, l level parameter, B is the bundle containing all information previously generated, λ regulates l update, *best_l* boolean to choose between arbitrary l (*false*) or the best value v^i (*true*), m regulates the choice between NS and SS.

Output the best μ which minimizes f and the current convex combinators θ.

function LBM($f, \bar{\mu}, l, B, \lambda, best\_l, m$)
  if *best_l* **then** ▷ Use of v^i for the LS, ignore l
    $(\mu, v) \leftarrow \arg\min\{v : v \geq \langle z^b, \mu \rangle - \alpha^b, b \in B\}$
    $l \leftarrow \lambda f(\bar{\mu}) + (1-\lambda)v$
    $\theta \leftarrow \arg\min\{\|\sum_{b \in B} z^b \theta^b\|_2^2/2 + \sum_{b \in B}(l + \alpha^b)\theta^b : \theta \geq 0\}$
  else ▷ Arbitrary l
    $empty\_MP \leftarrow true$
    while $empty\_MP$ **do**
      $\theta \leftarrow \arg\min\{\|\sum_{b \in B} z^b \theta^b\|_2^2/2 + \sum_{b \in B}(l + \alpha^b)\theta^b : \theta \geq 0\}$
      if θ unbounded **then** ▷ Check if the dual prob. is unbounded
        $l \leftarrow \lambda f(\bar{\mu}) + (1-\lambda)l$
      else
        $empty\_MP \leftarrow false$
      end if
    end while
  end if
  $d \leftarrow -\sum_{b \in B} z^b \theta^b$
  $v \leftarrow \max\left\{\langle z^b, \bar{\mu} + d \rangle - \alpha^b : b \in \mathcal{B}\right\}$
  if $f(\bar{\mu} + d) - f(\bar{\mu}) \leq m(v - f(\bar{\mu}))$ **then**
    $\bar{\mu} \leftarrow \bar{\mu} + d$ ▷ SS
  else
    $\bar{\mu} \leftarrow \bar{\mu}$ ▷ NS
  end if
  $z \leftarrow \sum_{b \in B} z^b \theta^b$
  $\alpha \leftarrow \sum_{b \in B} \alpha^b \theta^b$
  $B \leftarrow B.append(z, \alpha)$
  return $\bar{\mu}, \theta$
end function

Algorithm 2 implements the function to find a feasible x given a $\bar{\mu}$ as described in section 2.1.3. It has no parameters that need a description, but uses parameters that have already been widely seen during the description of the problem.

Algorithm 2 Algorithm to find a feasible x given $\bar{\mu}$

Input Q, q, E, b the parameters used for the function $x^T Qx + qx + \mu(Ex - b)$ that has to be minimized, u is the vector of right box constraints, $\bar{\mu}$ is the current value of μ.

Output feasible x which solves the MCF problem given $\bar{\mu}$

function GET_BOXED_X$(Q, q, E, b, u, \bar{\mu})$
    $n \leftarrow size(Q)$
    $\bar{x} \leftarrow zeros(n)$
    for $i = 0;\ i < n;\ i++$ **do**
        if $Q_{ii} = 0$ **then**
            if $q_i 0 + \bar{\mu}(E_{:,i} 0 - b) < q_i u_i + \bar{\mu}(E_{:,i} u_i - b)$ **then**
                $\bar{x}_i \leftarrow 0$
            else
                $\bar{x}_i \leftarrow u_i$
            end if
        else
            $x_i^* \leftarrow \frac{1}{2Q_{ii}}(-q_i - E_{i,:}^T \bar{\mu})$
            $\bar{x}_i \leftarrow \max(0, \min(x_i^*, u_i))$
        end if
    end for
    return $\bar{x}$
end function

Algorithm 3 implements the function to solve the MCF problem 1. It uses inside the LBM algorithm, so it has the same parameters plus the accuracy used for the for the stopping criterion of the algorithm (ϵ).

Algorithm 3 Algorithm to solve MCF problem 1 using LBM

Input Q diagonal, positive semidefinite matrix $\in \mathbb{R}^{n\times n}$, q vector $\in \mathbb{R}^n$, E matrix of node-arc directed connected graph $\in \mathbb{R}^{m\times n}$, b vector $\in \mathbb{R}^m$, u vector of right box constraints $\in \mathbb{R}^n$, ϵ accuracy in the stopping criterion, l level parameter, λ regulates l update, *best_l* boolean to choose between arbitrary l (*false*) or the best value v^i (*true*), m regulates the choice between NS and SS.
Output the best x which minimizes the MCF problem.

function QMCF_SOLVER$(Q, q, E, b, u, \epsilon, l, \lambda, best_l, m)$
    $m, n \leftarrow size(E)$
    $x\_satisfying\_const\_found \leftarrow False$
    $B \leftarrow [\,]$
    $X \leftarrow [\,]$
    $f(x) \leftarrow x^T Qx + qx$ ▷ Parametric function in x
    $L(x, \mu) \leftarrow x^T Qx + qx + \mu(Ex - b)$ ▷ Parametric function in x and μ
    $\bar{\mu} \leftarrow 0$ ▷ μ computed in each iteration
    $\bar{x} \leftarrow get\_boxed\_x(Q, q, E, b, u, \bar{\mu})$ ▷ x computed in each iteration
    $x\_best \leftarrow \emptyset$ ▷ x feasible giving the best Upper Bound
    $Dual\_f(\mu) \leftarrow -\bar{x}^T Q\bar{x} - q\bar{x} - \mu(E\bar{x} - b)$ ▷ Parametric function in μ
    $z \leftarrow E\bar{x} - b$
    $\alpha \leftarrow \langle z, \bar{\mu}\rangle - Dual\_f(\bar{\mu})$
    $B \leftarrow B.append(z, \alpha)$
    $LB \leftarrow L(\bar{x}, \bar{\mu})$ ▷ Lower Bound
    $UB \leftarrow f(\bar{x})$ ▷ Upper Bound
    while $|UB - LB| \geq \epsilon$ **do**
        $\bar{\mu}, \theta \leftarrow LBM(Dual\_f(\mu), \bar{\mu}, l, B, \lambda, best\_l, m)$
        if $\sum_i \theta^i = 1 \wedge \sum_{b\in B} z^b\theta^b = 0$ **then**
            $\bar{x} \leftarrow \sum_i x^i\theta^i$ ▷ $x^i \in X$
            $x\_satisfying\_const\_found \leftarrow True$
        else
            if $\sum_i \theta^i \neq 1$ **then**
                $\theta \leftarrow \frac{\theta}{\langle\theta,\, e\rangle}$
            end if
            $\bar{x} \leftarrow \sum_i x^i\theta^i$
            $\bar{x} \leftarrow \bar{x} - E^+(E\bar{x} - b)$ ▷ $\arg\min \left\{\frac{1}{2}\|x - \bar{x}\|_2^2 : Ex = b\right\}$
            if $\bar{x} \in [0, u]$ **then**
                $x\_satisfying\_const\_found \leftarrow True$
            else
                $\bar{x} \leftarrow get\_boxed\_x(Q, q, E, b, u, \bar{\mu})$
                if $x\_best = \emptyset$ **then**
                    $UB = f(\bar{x})$
                end if
            end if
        end if

```
        X ← X.append(x̄)
        if x_satisfying_const_found = True then
            if f(x̄) < f(x_best) then
                x_best ← x̄
                l ← f(x_best)
                UB ← f(x_best)
            end if
            x_satisfying_const_found ← False
        end if
        if L(x̄, μ̄) > LB then
            LB ← L(x̄, μ̄)
        end if
        Dual_f(μ) ← −x̄^T Q x̄ − q x̄ − μ(E x̄ − b)
        z ← B[−1].z                          ▷ Get last z from the bundle
        α ← B[−1].α                          ▷ Get last α from the bundle
    end while
    return x_best
end function
```

3.2 Convergence

In the following, we analyze the convergence of this algorithm. First of all, we can guarantee that the bundle method with the use of a level parameter is convergent. To prove this, we recall Theorem 2.3.3 from chapter 15 of *Convex Analysis and Minimization Algorithms II*[1]. The proof of this Theorem is not reported here, but it states that if we know in advance the optimal value x_* and its respective $f(x_*)$ that allows you to define the level parameter $l^i = f(x_*)$, then the LBM generates a minimizing sequence $\{x_i\}$. Unluckily, this is not our case since we do not have any optimal solution in advance. Despite this, as described in Remark 2.3.4[1], a minimizing sequence can be found even if the optimal value is not known in advance, so enforcing the definition of the level parameter without the use of the optimal value (so updating l^i with the use of v^i as we've seen before: $l^{i+1} = \lambda f(\bar{\mu}^i) + (1 - \lambda)v^i$, since v^i is constantly the best approximation for the optimal $f(x_*)$). This is a good result, but how do we know when we have arrived at an optimal solution?

Thanks to the primal function structure, we can exploit some values computed in Algorithm 3 to bound the optimal value $v(P)$. The value $L(\bar{x}_i, \mu_i)$ is a lower bound for $v(P)$, while the value $f(\bar{x}_i)$ is an upper bound for $v(P)$.

Note that, in order to have always the best upper bound found so far, we only evaluate $f(\bar{x}_i)$ using an $\bar{x}_i$ computed through the cases 1 or 2 described in 2.2.3, since they always return a feasible solution.

Regarding the lower bound, it's evaluated on the pair $\bar{x}_i$, μ_i. Here μ_i is the best μ value found so far for the function L, while $\bar{x}_i$ can be obtained through any method $(1, 2$ or $3)$. This allows us to get an available estimate of the gap between the obtained value $f(\bar{x}_i)$ and the optimal value $v(P)$ for the Primal problem. Indeed we

[1] Convex Analysis and Minimization Algorithms II. `https://link.springer.com/book/10.1007/978-3-662-06409-2`

can state that:

$$f(\bar{x}_i) - v(P) \leq f(\bar{x}_i) - L(x_i, \mu_i) \tag{30}$$

This estimation of the gap is used as a stopping condition in Algorithm 3, where we iterate until $|f(\bar{x}) - L(\bar{x}, \bar{\mu})| \geq \epsilon$. So we compare the distance between the best Lower bound and the best Upper bound so far. This is because when the condition is no longer satisfied, then the value $f(\bar{x}^i)$ is "close enough" to the optimal value $v(P)$.

4 Experiments