

Globally convergent decomposition algorithm for risk parity problem in portfolio selection

A.Cassioli, G.Cocchi, F.D'Amato, M.Sciandrone

February 15, 2016

1 Introduction

2 Preliminary background

Let us consider the following optimization problem:

$$\min_{x,y} f(x,y) \tag{1a}$$

$$\text{s.t. } l \leq x \leq u \tag{1b}$$

$$\mathbf{1}^T x = 1 \tag{1c}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$, f continuously differentiable, $l, u \in \mathbb{R}^n$ with $l < u$ and $\mathbf{1} \in \mathbb{R}^n$ is all composed by ones.

Now we define the feasible set \mathcal{F} of Problem (1):

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^{n+k} : \mathbf{1}^T x = 1, l \leq x \leq u\}. \tag{2}$$

Since the constraints of Problem (1) respect constraints qualification conditions, a feasible point (x, y) is a stationary point, if the Karush-Kuhn-Tucker (KKT) conditions are satisfied.

Let $L(x, y, \lambda, \mu, \gamma)$ the Lagrangian function associated to Problem (1) then we can write KKT conditions.

Proposition 2.1 (Optimality conditions (Necessary)) *Let $(x^*, y^*) \in \mathbb{R}^{n+k}$, with $(x^*, y^*) \in \mathcal{F}$, a local optimum for Problem (1). Then there exist three multipliers $\lambda^* \in \mathbb{R}^n$, $\mu^* \in \mathbb{R}^n$, $\gamma^* \in \mathbb{R}$ such that:*

$$\begin{aligned} \nabla_x L(x^*, y^*, \lambda^*, \mu^*, \gamma^*) &= \nabla_x F(x^*, y^*) + \lambda^* - \mu^* + \gamma^* = 0 \\ \nabla_x L(x^*, y^*, \lambda^*, \mu^*, \gamma^*) &= \nabla_y F(x^*, y^*) = 0 \\ \lambda_i^* (l - x_i^*) &= 0, \quad \forall i \\ \mu_i^* (x_i^* - u) &= 0, \quad \forall i \\ \lambda^*, \mu^* &\geq 0 \end{aligned} \tag{3}$$

From the first condition we have:

$$\nabla_x f(x^*, y^*) - \lambda^* + \mu^* + \gamma^* = 0 \quad (4)$$

Then there are three possible cases:

$$\frac{\partial F(x^*, y^*)}{dx_i} = \begin{cases} -\mu_i^* - \gamma & x_i^* = u \\ -\gamma + \lambda_i^* & x_i^* = l \\ -\gamma & l < x_i^* < u \end{cases} \quad (5)$$

Then if $x_j^* > l$:

$$\frac{\partial f(x^*)}{dx_j} \leq \frac{\partial f(x^*)}{dx_i}, \forall i \quad (6)$$

Let $(x, y) \in \mathcal{F}$, we define a set of all feasible direction in (x, y) :

$$\mathcal{D}(x, y) = \{d \in \mathbb{R}^{n+k} : \mathbf{1}^T d_x = 0, d_i \geq 0 \forall i \in L(x), d_i \leq 0 \forall i \in U(x)\} \quad (7)$$

where:

$$\begin{aligned} L(x) &= \{i : x_i = l\} \\ U(x) &= \{i : x_i = u\} \end{aligned} \quad (8)$$

2.1 Set of sparse feasible directions

Because of y is unconstrained, we pay attention only on the x variable to find a feasible descent direction w.r.t. x . In our case we want to build a set of sparse feasible direction in order to justify our decomposition approach.

Let $(x, y) \in \mathcal{F}$ non stationary w.r.t. x , then it's easy to see that

$$L(x) \neq \{1, \dots, n\} \quad (9)$$

hence $\exists i$ such that $x_i > 0$ and $j \neq i$ such that:

$$\frac{\partial f(x^*)}{dx_i} > \frac{\partial f(x^*)}{dx_j}, \quad (10)$$

Now we define a direction $d^{i,j} \in \mathbb{R}^n$ with only two non-zero components such that:

$$d_h^{i,j} = \begin{cases} 1, & h = j \\ -1, & h = i \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Proposition 2.2 *Let (x, y) a feasible point for Problem (1). Then the direction $d^{i,j}$ is feasible and descent direction in x .*

Proof For the feasibility it is enough to see that $\mathbf{1}^T d^{i,j} = 1 - 1 = 0$. Then we can apply sufficient conditions for descent direction in x , such that:

$$\nabla_x f(x, y)^T d^{i,j} = \frac{\partial f(x)}{dx_j} - \frac{\partial f(x)}{dx_i} < 0;$$

As always, we should choose the steepest descent direction composed by only two non-zero components. This can be done computing the *Most Violating Pair* (i, j) such that $x_i > 0$ and:

$$(i, j) \in \arg \min_{l, m} \left\{ \frac{\partial f(x^*)}{\partial x_l} - \frac{\partial f(x^*)}{\partial x_m} \right\} \quad (12)$$

If one doesn't want to use decomposition methods, he can define a direction $d_{xy} \in \mathbb{R}^{n+k}$ such that:

$$d_{xy} = \{d^{i,j}, -\nabla_y f(x, y)\} \quad (13)$$

2.2 Armijo-Type Line Search Algorithm

In this section, we briefly describe the well-known Armijo-type line search along a feasible descent direction. The procedure will be used in the decomposition method presented in the next section. Let $d^k \in \mathcal{D}(x_k)$ $x^k \in \mathcal{F}$. In particular we choose $d^k = d_k^{i,j}$ with MVP $(i(k), j(k))$. We denote by Δ_k the maximum feasible step along d^k .

It is easy to see that:

$$\Delta_k = \min\{x_{j(k)}^k - l, u - x_{i(k)}^k\}$$

Then at iteration $k + 1$ we have:

$$x_{j(k)}^{k+1} = \begin{cases} l & \alpha_k = x_{j(k)}^k - l \\ x_{j(k)}^k - u + x_{i(k)}^k & \alpha_k = u - x_{i(k)}^k \end{cases}$$

and:

$$x_{i(k)}^{k+1} = \begin{cases} x_{j(k)}^k - l + x_{i(k)}^k & \alpha_k = x_{j(k)}^k - l \\ u & \alpha_k = u - x_{i(k)}^k \end{cases}$$

2.3 Exact Line Search

When we move along the direction $d^{i(k), j(k)}$, defined in (11), we modify only 2 variables $(x_{i(k)}, x_{j(k)})$ leaving the others unchanged. Thus, we can see our $f(x, y)$ as a function of two components, i.e. we can rewrite Problem (1) as

$$\min_{x_{i(k)}, x_{j(k)}} f(x_{i(k)}, x_{j(k)}) \quad (14a)$$

$$\text{s.t. } l_{i(k)} \leq x_{i(k)} \leq u_{i(k)} \quad (14b)$$

$$l_{j(k)} \leq x_{j(k)} \leq u_{j(k)} \quad (14c)$$

$$x_{i(k)} + x_{j(k)} = 1 - \underbrace{\sum_{h \neq i(k), j(k)} x_h}_c \quad (14d)$$

Thanks to the last constraint, we can substitute $x_{i^*} = c - x_{j^*}$ and then we obtain

$$\min_{x_{j(k)}} f(x_{j(k)}) \quad (15a)$$

$$\text{s.t. } x_{i(k)} = c - x_{j(k)} \quad (15b)$$

$$l_{j(k)} = \max\{l_{j(k)}, c - u_{i(k)}\} \leq x_{j(k)} \leq \min\{u_{j(k)}, c - l_{i(k)}\} = uu_{j(k)} \quad (15c)$$

Because the domain is $I = [l_{j(k)}, uu_{j(k)}]$, and $f(x_{j(k)})$ is continuous and differentiable in I , then f has a minimum in I and we can compute $f'(x_{j(k)})$. Let $R = \{r \mid f'(r) = 0, r \in I\}$ be set of the real feasible roots of f' . Each $r \in R$ can be a local maximum, minimum or flex; if $R = \{\emptyset\}$, then the minimum of f is on the extreme points of I .

Let $r^* = \arg \min_{r \in R} f(r)$, then the optimal step α^* along the direction $d^{i(k), j(k)}$ is

$$\alpha^* = x_{j(k)} - r^* > 0 \quad (16)$$

TODO: problema di notazione, $x_{j(k)}$ rappresenta il valore della componente $j(k)$ del vettore x prima della ricerca di linea

3 A decomposition framework

The algorithm follows the Gauss-Seidel scheme with 2 blocks of variables (x and y). At each iteration, we optimize f w.r.t. one block of variables, considering the other block fixed.

The y -block of variables is unconstrained, so we can find

$$y^{(k+1)} = \arg \min_{y \in \mathbb{R}^m} f(x^{(k)}, y) \quad (17)$$

using the first-order necessary condition, i.e. *TODO: ma questo vale se f è convessa rispetto ad y*

$$\nabla_y f(x^{(k)}, y^{(k)}) = 0 \quad (18)$$

4 Convergence analysis

Proposition 4.1 *Let $\{(x^k, y^k)\}$ be the sequence of points generated by the decomposition algorithm. Then*

- $(x^k, y^k) \in \mathcal{L}_0 \ \forall k$
- $\{(x^k, y^k)\}$ admits limit points and each limit point is a solution for Problem (1)

Algorithm 1: Decomposition Algorithm

Data: Given the initial feasible guess $(x^{(0)}, y^{(0)})$

- 1 Set $k = 0$
- 2 **while** (*not convergence*) **do**
- 3 Compute $y^{(k+1)}$ as in (17)
- 4 Compute $\nabla_x f(x^{(k)}, y^{(k+1)})$
- 5 Choose indexes $i(k), j(k)$ using the Gauss-Southwell rules
- 6 Choose a step $\alpha^{(k)}$ along the direction $d^{i(k), j(k)}$
- 7 Set $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{i(k), j(k)}$
- 8 Set $k = k + 1$

Proof From (17) we have

$$f(x^{(k)}, y^{(k+1)}) \leq f(x^{(k)}, y^{(k)}) \quad (19)$$

and from the update rule of x it follows (thanks to the Armijo properties)

$$f(x^{(k+1)}, y^{(k+1)}) \leq f(x^{(k)}, y^{(k+1)}) \quad (20)$$

Finally we have

$$f(x^{(k+1)}, y^{(k+1)}) \leq f(x^{(k)}, y^{(k)}) \quad (21)$$

Let (x^*, y^*) be a limit point of $\{(x^k, y^k)\}$, i.e. there exist an infinite subset $K \subseteq N$ such that

$$\lim_{k \in K, k \rightarrow \infty} (x^k, y^k) = (x^*, y^*) \quad (22)$$

By contradiction, let us assume that (x^*, y^*) is not a solution. In this case, at least one of the following conditions hold:

$$\nabla_y f(x^*, y^*) \neq 0 \quad (23a)$$

$$\exists i, j \text{ s.t. } x_i^* > l_i \text{ and } \nabla_x f(x^*, y^*)^T d^{i,j} < 0 \quad (23b)$$

For Point (1), we recall that for (17) $y^{(k+1)}$ minimizes $f(x^{(k)}, y)$ with respect to y . So, if we imagine to perform Armijo along a descent direction (for example, $-\nabla_y f(x^{(k)}, y^{(k)})$) we have

$$f(x^{(k)}, y^{(k+1)}) \leq f(x^{(k)}, y^{(k)}) - \gamma \alpha \|\nabla_y f(x^{(k)}, y^{(k)})\|^2 \quad (24)$$

Thanks to (20) we can extend the Armijo convergence properties to $(x^{(k+1)}, y^{(k+1)})$. In particular, for Armijo we can write

$$\lim_{k \rightarrow \infty} \|\nabla_y f(x^{(k)}, y^{(k)})\| = \|\nabla_y f(x^*, y^*)\| = 0 \quad (25)$$

So we have demonstrate that Point (1) cannot hold.

For Point (2), the algorithm performs an Armijo line search along $d^{i(k), j(k)}$ to

determine the step $\alpha^{(k)}$. Thanks to Armijo we can write

$$\lim_{k \rightarrow \infty} \alpha^{(k)} \|d^{i(k),j(k)}\| \frac{|\nabla_x f(x^{(k)}, y^{(k+1)})^T d^{i(k),j(k)}|}{\|d^{i(k),j(k)}\|} = 0 \quad (26)$$

From (11), we know that $\|d^{i(k),j(k)}\| = \sqrt{2}$, so from (26) it follows

$$\lim_{k \rightarrow \infty} \alpha^{(k)} \left| \nabla_x f(x^{(k)}, y^{(k+1)})^T d^{i(k),j(k)} \right| = 0 \quad (27)$$

So, at least one of the following conditions hold:

1. $\lim_{k \rightarrow \infty} \alpha^{(k)} = 0$
2. $\lim_{k \rightarrow \infty} \left| \nabla_x f(x^{(k)}, y^{(k+1)})^T d^{i(k),j(k)} \right| = 0$

5 A case study: Portfolio Selection

In this section, we show that the general framework presented above can be used to a specific class of portfolio selection problem, namely the Risk Parity portfolio selection. Let $x \in \mathbb{R}^n$ be the portfolio, where x_i represents the fraction of budget that is invested in the asset i , and let Q be the covariance matrix between the assets; in [?] is introduced the following least-square optimization problem:

$$\min_{x, \theta} f(x, \theta) = \sum_{i=1}^n (x_i q_i^T \cdot x - \theta)^2 \quad (28a)$$

$$\text{s.t. } l \leq x \leq u \quad (28b)$$

$$\mathbf{1}^T x = 1 \quad (28c)$$

where $\theta \in \mathbb{R}$ and q_i is the i -th column of Q . Note that the problem is not necessary convex with respect to x but it is strictly convex and coercive with respect to θ . Infact we have:

$$\frac{\partial^2}{\partial^2 \theta} f(x, \theta) = 2 \quad (29)$$

which is a sufficient condition for a strictly convex function. At each iteration k , we have

$$\theta^{(k+1)} = \arg \min_{\theta} f(x^{(k)}, \theta) \quad (30)$$

f is strictly convex with respect to θ , so we can find $\theta^{(k+1)}$ as a solution of

$$\nabla_{\theta} f(x^{(k)}, \theta) = 0 \quad (31)$$

That is

$$\theta^{(k+1)} = \frac{\sum_{i=1}^n x_i^{(k)} q_i^T \cdot x^{(k)}}{n} \quad (32)$$

| n | G-S _(LS) | G-S _(OPT) | SNOPT |
|------|---------------------|----------------------|--------|
| 5 | 100.0% | 100.0% | 100.0% |
| 10 | 100.0% | 100.0% | 100.0% |
| 20 | 100.0% | 100.0% | 100.0% |
| 50 | 99.0% | 99.4% | 99.5% |
| 100 | 85.4% | 87.0% | 91.2% |
| 200 | 40.0% | 45.0% | 52.0% |
| 300 | 22.0% | 24.0% | 34.0% |
| 500 | 0.0% | 6.0% | 3.0% |
| 750 | 0.0% | 2.0% | 0.0% |
| 1000 | 0.0% | 0.0% | 0.0% |
| 1250 | 0.0% | 0.0% | 0.0% |

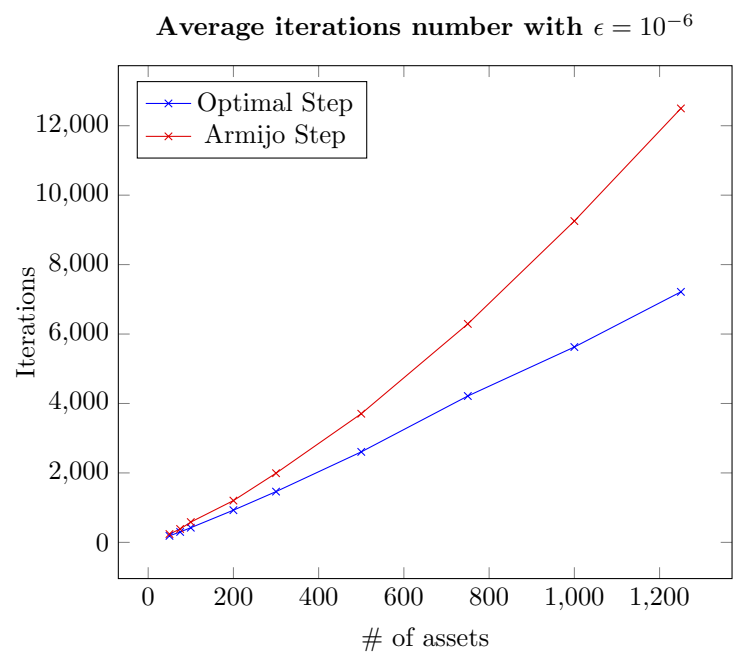
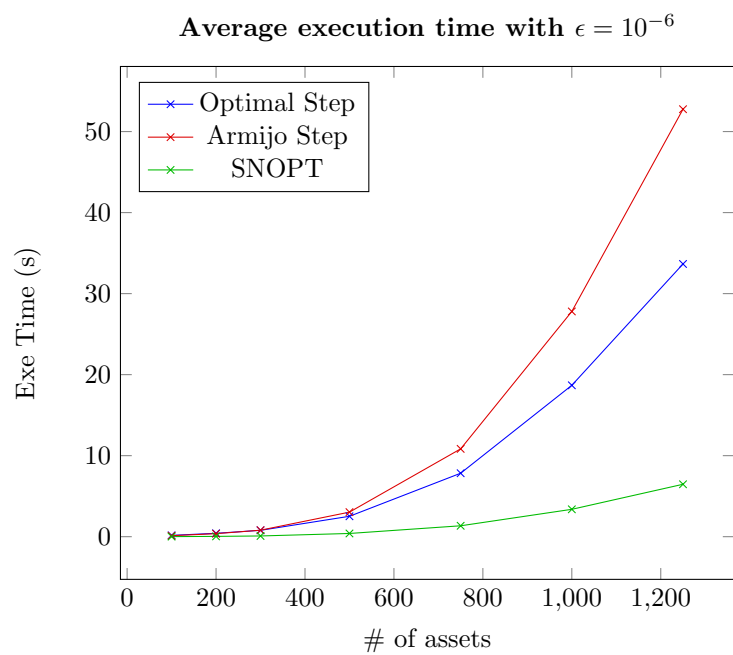
Table 1: $x_i^{(0)} = \frac{1}{n} \quad \forall i$

| n | G-S _(LS) | G-S _(OPT) | SNOPT |
|------|---------------------|----------------------|--------|
| 5 | 100.0% | 100.0% | 100.0% |
| 10 | 100.0% | 100.0% | 100.0% |
| 20 | 99.8% | 99.8% | 99.8% |
| 50 | 100.0% | 99.8% | 99.1% |
| 100 | 100.0% | 99.8% | 96.6% |
| 200 | 100.0% | 99.6% | 94.8% |
| 300 | 100.0% | 99.0% | 92.0% |
| 500 | 100.0% | 99.0% | 92.5% |
| 750 | 99.0% | 98.0% | 86.5% |
| 1000 | 96.0% | 95.0% | 76.0% |
| 1250 | 90.0% | 93.0% | 76.5% |

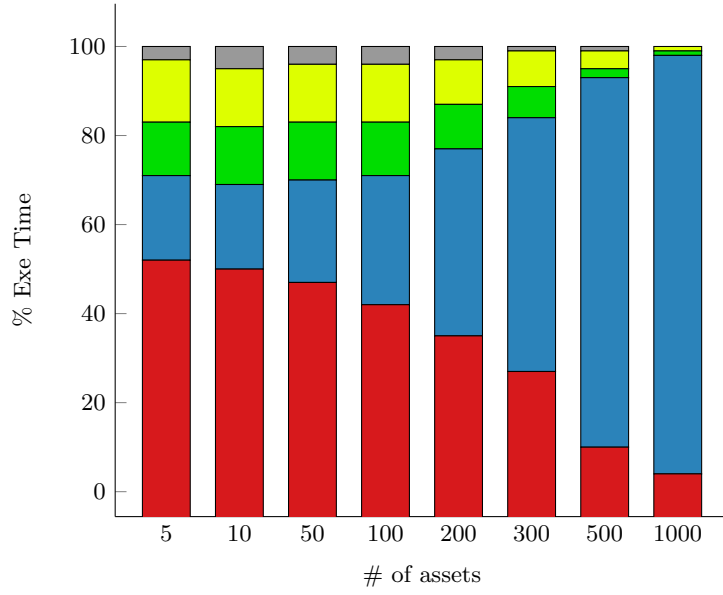
Table 2: $x^{(0)} = [1, 0, \dots, 0]$

6 Computational experiments

We compare the convergence properties of the algorithms in term of number of times that one algorithm converges to a stationary point that is the minimum point for the convex formulation.



Armijo line search



Optimal step computation

