noML09: Deflected Subgradient Methods for a Dual Formulation of Convex Quadratic Separable Min Cost Flow Boxed Problems

The Problem and The Approach

We consider a dual approach to solve the Min Cost Flow box constrained convex quadratic separable problem (P) defined by

$$\nu_*(P) \coloneqq \inf_{x \in S_{(P)}} \frac{1}{2} x^\intercal Q x + q^\intercal x$$
$$\text{where} \qquad S_{(P)} \coloneqq \{x \,:\, Ex = b \;\wedge\; l \le x \le u\}$$

where, here and in the following, E, of size (m, n), is the node-arc incidence matrix of a directed graph, $Q \succcurlyeq 0$ is a diagonal matrix; we also let $f : \mathbb{R}^n \to \mathbb{R}$ denote the quadratic polynomial to be minimized, $f(x) \coloneqq \frac{1}{2} x^\intercal Q x + q^\intercal x$.
We choose one of the simplest dual reformulations and approach its iterative solution implementing some subgradient methods; hence we touch upon deflected methods, automatic parameter tuning, heuristics and one of the "well known" properties of subgradient methods in dual reformulations.

Notation ▶ With subspace we mean a subset of $\mathbb{R}^k$, for some $k \in \mathbb{N}$. Comparisons between vectors in $\mathbb{R}^n$ are to be intended coordinate-wise, in the canonical basis; in the same spirit, the box described by $\{x : l \le x \le u\}$ is simply written as $[l, u]$. The linear subspace described by $\{x : Ex = b\}$ is named $\mathcal{H}_{E,b}$. $\|\cdot\|_F$ is the Frobenius norm defined as $\|A\|_F^2 \coloneqq \operatorname{Tr} A^\intercal A = \sum_i \sigma_i^2$, where σ_i are the singular values of A. Functions are polymorphic, however the meaning should be evident from the context. Functions with a clear meaning for scalar values, such as max and min, when applied to vectors entail a broadcast, e.g. let u, v be two vectors in $\mathbb{R}^k$, then $\max(u, v) \in \mathbb{R}^k$ and $\max(u, v)_i = \max(u_i, v_i)$. Similarly, for any scalar q, $\max(u, q) \in \mathbb{R}^k$ and $\max(u, q)_i = \max(u_i, q)$. We conveniently define $\inf_\emptyset \cdot \coloneqq +\infty$; analogous definition holds for $\sup_\emptyset$ et similia. We will stick to the *subgradient* appellative also when functions are concave.

1. The Dual Reformulation

The objective function in (P) is quadratic convex, the constraints are affine; box constraints can be effectively kept implicitly in the domain of definition of the Lagrangian. Strong duality holds, with the caveat that the optimal duality gap is to be considered null when both the optimal primal and optimal dual values are $+\infty$. This is a standard results and can be seen considering that constraints are affine and, in the box, the objective function is quadratic convex bounded so there exists a non-vertical separating hyperplane whenever $S_{(P)} \neq \emptyset$.

(D1) Dual Flux Conservation Constraints ▶ We relax the flux conservation constraints:

$$\nu_*(P) = \inf_{x \in S_{(D1)}} \sup_{\mu} L(x, \mu) \quad \ge \quad \sup_{\mu} \inf_{x \in S_{(D1)}} L(x, \mu) =: \nu^*(D1)$$
$$\text{where} \qquad L(x, \mu) \coloneqq \frac{1}{2} x^\intercal Q x + q^\intercal x + \mu^\intercal (Ex - b)$$
$$\text{and} \qquad S_{(D1)} \coloneqq [l, u].$$

The Lagrangian dual is defined by

$$L(\mu) \coloneqq \inf_{x \in S_{(D1)}} L(x, \mu);$$

concurrently, the parametric set describing the relaxed primal points corresponding to a given dual point is defined as

$$X(\mu) \coloneqq \arg \inf_{x \in S_{(D1)}} L(x, \mu). \tag{1}$$

Since the feasible space of the dual problem is a box and the Lagrangian is separable as a function of x for a fixed μ, $X(\mu)$ can be described component-wise[1], as the Cartesian product $X(\mu) = \prod X(\mu)_j$. Consider then the j-th component. If $Q_{jj} > 0$, thanks to convexity, it is sufficient to solve for x_j in

$$\frac{\partial L(x, \mu)}{\partial x_j} = (Qx + q + E^\intercal \mu)_j = 0$$

and then project to the nearest side of the box:

$$X(\mu)_j = \{\max(l_j, \min(u_j, -Q_{jj}^{-1}(q + E^\intercal \mu)_j))\}.$$

Else, if $Q_{jj} = 0$,

[1] In the same canonical basis, in which Q is diagonal

$$X(\mu)_j = \arg\min_{x_j\in[l_j,u_j]} (q+E^\intercal\mu)_j x_j = \begin{cases} \{l_j\} & \text{if } (q+E^\intercal\mu)_j > 0 \\ \{u_j\} & \text{if } (q+E^\intercal\mu)_j < 0 \\ [l_j,u_j] & \text{otherwise.} \end{cases}$$

Any specific value of $x(\mu) \in X(\mu)$ that we choose corresponds to the choice of a subgradient in the subdifferential

$$\partial L(\mu) = \{Ex - b : x \in X(\mu)\}; \tag{2}$$

without further specification, $x(\mu)$ represents an $x \in X(\mu)$. Clearly $L(\mu)$ is "piecewise differentiable" and

$$\forall g \in \partial L(\mu), \quad \|g\| \leq \|E\| (\|u\| + \|l\|) + \|b\|. \tag{3}$$

2. The Feasible Polyhedron

The feasible space $S_{(P)}$ of the primal problem is the intersection of the box $[l,u]$ with the linear subspace $\mathcal{H}_{E,b}$ described by flux conservation constraints, $Ex = b$. In order to devise a method to recast information we gain from calculations in the dual space to the primal space, some form of projection from $S_{(D1)}$ onto $S_{(P)}$[2] will be required. In the following paragraphs of this section we start with a short detour on basic properties of the incidence matrix E, to arrive to develop the theory to justify our implementation choices.

About E: bounding the distance from $S_{(P)}$ ► The first well known property of the incidence matrix E of a graph G is that, if the graph G is connected and it's not a singleton, the left kernel of E is generated by the vector of all 1s. Select the submatrix $E(T) \subseteq E$ corresponding to a spanning tree T of G. Then remove any node r, that is a row of $E(T)$; such node is conveniently designated as root of the tree. In this manner we obtained an invertible matrix that we call $\hat{E}(T_r)$. It's straightforward to calculate $\hat{E}(T_r)^{-1}$ with a simple analysis of its left action on $\hat{E}(T_r)$: consider the row i of $\hat{E}(T_r)^{-1}$; the result of its left action on $\hat{E}(T_r)$ is a linear combination of the rows of $E(T)$, and it should add to e_i, the row vector with a 1 at the ith place. There are two rows of $E(T)$ which are nonzero at column i, pick the one farther from the node r, with the appropriate sign to make it 1 at column i. If we continue, picking with the same sign the whole subtree which doesn't include r and rooted in the first vertex we selected, we'll obtain the whole i-th row of $\hat{E}(T_r)^{-1}$. In order to derive a bound independent from the particular topology of E, let T' range over all the possible trees with m nodes. Since each arc is appearing as a ± 1 in the matrix $\hat{E}(T'_r)^{-1}$ as many times as the number of its parents arcs in the tree, it's straightforward to determine greedily the trees leading to the sparsest and the densest $\hat{E}(T'_r)^{-1}$: the sparse one goes wide, e.g. is a star tree rooted in the center ($\hat{E}(T'_r)$ is the identity), the dense one, $T^d_r(m)$, goes far, e.g. is a path rooted at one of the extremal nodes ($\hat{E}(T^d_r(m))^{-1}$ is the upper triangular matrix filled with ± 1).
Two easy corollaries follow:

- It's more advantageous to calculate the left action of the inverse recursively, since each subtree can be considered as the appropriately union of its own proper subtrees and the root. As a consequence, given the sparse matrix $\hat{E}(T_r)$, calculating its action has the same O-cost as calculating the action of its inverse.
- $\hat{E}(T_r)^{-1}$ contains only 0, ± 1, hence, denoting with σ the singular values, $\sigma_{\max}(\hat{E}(T_r)^{-1}) \leq \left\|\hat{E}(T_r)^{-1}\right\|_F \leq \left\|\hat{E}\left(T^d_r(m)\right)^{-1}\right\|_F = \sqrt{\binom{m}{2}}$.

Now consider the linear subspace $\mathcal{H}_{E,b}$ containing our feasible set; suppose that $x \in \mathcal{H}_{E,b'}$, with $b' = b + \Delta b$: *how far is x from $\mathcal{H}_{E,b}$?*
We can bound the distance with a function of the *least nonzero* singular value $\sigma_{\min}(E)$ and Δb. To evaluate $\sigma_{\min}(E)$ we can restrict our attention to connected graphs - otherwise we take the minimum among the $\sigma_{\min}$ of each connected component.
We can derive a lower bound for $\sigma_{\min}(E)$ from the upper bound for $\sigma_{\max}\left(\hat{E}(T_r)^{-1}\right)$. Let $\mathfrak{S}^j_k$ be the set of subsets of $\{1,\ldots,k\}$ of cardinality j; for any $s \in \mathfrak{S}^j_k$ we can associate the canonical projection for Euclidean spaces $\mathbb{P}_s$. Let $V \subset \mathbb{R}^n$ be a linear subspace and restrict its dimension to be $m-1$; it holds that

$$\begin{aligned} \sigma_{\min}(E) = \sup_V \inf_{\substack{v\in V \\ \|v\|=1}} \|Ev\| \geq & \sup_{s\in\mathfrak{S}^{m-1}_n} \inf_{\substack{v\in\mathbb{P}_s\mathbb{R}^n \\ \|v\|=1}} \|Ev\| \geq \\ \geq & \sup_{\substack{s\in\mathfrak{S}^{m-1}_n \\ t\in\mathfrak{S}^{m-1}_m}} \inf_{\substack{v\in\mathbb{P}_s\mathbb{R}^n \\ \|v\|=1}} \|\mathbb{P}_t Ev\| \geq \\ \geq & \sup_{T_r} \inf_{\substack{u\in\mathbb{R}^{m-1} \\ \|u\|=1}} \left\|\hat{E}(T_r)u\right\| = \\ = & \sup_{T_r} \sigma_{\min}\left(\hat{E}(T_r)\right) = \end{aligned}$$

[2]or on a space of analogous description

$$= \frac{1}{\inf_{T_r}\ \sigma_{\max}\left(\hat{E}(T_r)^{-1}\right)} \geq \frac{\sqrt{2}}{\sqrt{m(m-1)}}.$$

The following corollary is then immediate.

Corollary 0.1 Distance of $\mathcal{H}_{E,b+\Delta b}$ from $\mathcal{H}_{E,b}$. *The distance of $\mathcal{H}_{E,\Delta b}$ from the origin is bounded by $\sqrt{\binom{m}{2}}\,\|\Delta b\|$.*

Proof: With a singular value decomposition, we can write $E = \sum_{i=1}^{\mathrm{rk}(E)} \sigma_i v_i u_i^\intercal$, $\sigma_i > 0$; if we let $x = \arg\min_{Ex'=\Delta b} \|x'\|$, then $\forall y \in \ker E,\ x \cdot y = 0$, hence it holds that $x = \sum_i u_i u_i \cdot x$. Since from $Ex = \Delta b$ we obtain $\sigma_i x \cdot u_i = \Delta b \cdot v_i$, we conclude with

$$\|x\| = \left\| \sum_i u_i \frac{\Delta b \cdot v_i}{\sigma_i} \right\| \leq \frac{\|\Delta b\|}{\sigma_{\min}} \leq \sqrt{\binom{m}{2}}\,\|\Delta b\|\,. \qquad \square$$

If the starting point was sufficiently inside the box $[l,u]$, e.g. at a distance greater than $\sqrt{\binom{m}{2}}\,\|\Delta b\|$ from the boundary, the aforementioned upper bound is valid also for the distance from the feasible set $S_{(P)}$.
Otherwise, from the amount of flow that a max-flow algorithm would move to satisfy the required flux conservation constraints, the distance from the feasible space is upper bounded by $n\,\|\Delta b\|_1 \leq n\sqrt{m}\,\|\Delta b\|$ [3]. Thus we have shown the following:

Proposition 0.2 Distance from $S_{(P)}$: bound. *Given an $x \in \mathcal{H}_{E,b'} \cap [l,u]$, the following upper bound for the distance of x from the feasible space $S_{(P)} \neq \emptyset$ holds:*

$$d\left(x, S_{(P)}\right) \leq \begin{cases} \sqrt{\binom{m}{2}}\,\|\Delta b\| & \text{if } d(x, \partial[l,u]) \geq \sqrt{\binom{m}{2}}\,\|\Delta b\| \\ n\,\|\Delta b\|_1 \leq n\sqrt{m}\,\|\Delta b\| & \text{otherwise.} \end{cases}$$

Spanning trees and cycles: $\ker E$ ▷ For the sake of completeness in the combinatorial interpretation of the geometry, we show how the choice of a tree T and a root r, to be removed, naturally leads to a choice of the basis for the circulation circuits of the graph. Let $\hat{E}(G \setminus T_r)$ be the incidence matrix of the graph G without the edges in T, where also the node r, and the respective row, has been removed. Consider the left action of $\hat{E}(T_r)^{-1}$ on a column of $\hat{E}(G \setminus T_r)$ corresponding to an arc $v_i \to v_j$ in $G \setminus T$, denoted e_{ij}. Each row of $\hat{E}(T_r)^{-1}$ corresponding to an arc in the unique path in T from the root r to a given node v_k, is nonzero at the column corresponding to the node v_k and vice versa, thus, taking care of the opposite signs, we deduce that the left action of $\hat{E}(T_r)^{-1}$ on the column of $\hat{E}(G \setminus T_r)$ corresponding to e_{ij} is the union of the two paths leaving from v_i and from v_j up to their lowest common ancestor in the rooted tree T. The two paths are taken with opposite signs and their union with the reversed arc e_{ij} is a circuit, which gives the combinatorial interpretation of the base of circuits described by the kernel basis $\begin{bmatrix} -\hat{E}(T_r)^{-1}\hat{E}(G \setminus T_r) \\ I \end{bmatrix}$.

Projecting on $S_{(P)}$ ▶ Proposition 0.2 gives upper bounds for the distance between an x living in the dual space $S_{(D1)}$ [4] and the primal space $S_{(P)}$. Given these theoretical results, how to actually project a given $x \in [l,u] \cap \mathcal{H}_{E,b+\Delta b}$ to the feasible polyhedron $[l,u] \cap \mathcal{H}_{E,b}$? We won't strictly need the Euclidean projection, but the common property which we require for any projection $\Pi : [l,u] \to S_{(P)}$ is

$$\lim_{x \to x^* \in S_{(P)}} \Pi x = x^*. \tag{Proj}$$

Exploiting such projections, we can obtain upper bounds to the optimal value $\nu_*(P)$ even if the iterates carried out in the dual space did not reach the optimum yet. The idea is to consider lower order approximations of the original problem, i.e. the 0th order approximation and the 1st order ones which are briefly explained in following paragraphs. For such models, performant combinatorial algorithms exist that open the way for a precise but cheaper projection onto the feasible space $S_{(P)}$.
Hereafter, property (Proj) is satisfied, as the variation of flow per arc remains bounded by $\|\Delta b\|_1$.
Order 0: max flow ▷ At order 0, there is no variable cost per edge; we can apply any max flow algorithm suited to a non-integer flux. Algorithms privileging shorter augmenting paths may grossly lead to a smaller variation in the objective value.
Order 1: shortest paths $\rightsquigarrow$ linear min cost flow ▷ We can assign a cost to each arc considering the gradient of the objective function or of the Lagrangian, then apply a linear min-cost max-flow algorithm. A relevant observation is that, for a given $x(\mu)$ satisfying equation (1), in exact arithmetic there is no negative cycle in the residual network. In fact, let $x(\mu) \in [l,u] \cap \mathcal{H}_{E,b'}$ and consider the case of linear costs described by $\nabla_x f(x(\mu))$; thus $x(\mu)$ is an optimal point of the problem (P'), that is the same as (P) except for the flux conservation constraints. Suppose there is a direction $d \in \ker E$ such that $d^\intercal \nabla f(x(\mu)) < 0$; since $x(\mu)$ is optimal for (P'), it must hold that $\forall \epsilon > 0,\ x + \epsilon d \notin [l,u]$. Analogous reasoning applies when the cost is described by $\nabla_x L(x(\mu), \mu)$.

[3] From Holder inequality, $\|x\|_p = \left(\sum_{i=1}^n x_i^p \cdot 1\right)^{\frac{1}{p}} \leq n^{\frac{1}{p}-\frac{1}{q}}\,\|x\|_q$.
[4] i.e. $x \in [l,u] \cap \mathcal{H}_{E,b+\Delta b}$ for some Δb.

The box who missed $\mathcal{H}_{E,b}$: nearest point in the box ▶ Suppose we are given a box $[l', u']$ and a linear subspace $\mathcal{H}_{E,b}$ whose reciprocal intersection is empty; consider the problem to determine the point in the box that is nearest to the linear subspace, with respect to the Euclidean norm. From a geometric point of view, we have iterative approaches from the corresponding quadratic box constrained minimization problem (`MQBProblem` in the package):

$$\arg \min_{x \in [l', u']} \|Ex - b\|^2.$$

Any of the projected conjugate gradient iterations, though agnostic of the peculiar form of E, is suited to the problem.

Effective upper bounds to $\nu_*(P)$ ▶ When $Q \succ 0$ even a simple max-flow projection is efficient, as shown in [11]; instead, when Q is singular, it's generally impossible to define a continuous $x : \mu \mapsto x(\mu) \in X(\mu)$. In particular, as $\mu \to \mu^*$, where $\mu^* \in \arg\sup L(\mu)$, there's no guarantee that $x(\mu) \to x^*$, where $x^* \in \arg\inf_{x \in S_{(P)}} f(x)$ is a feasible primal optimal point of (P). Still, from equation (2), it does hold that

$$x^* \in \arg \min_{x \in X(\mu^*)} \|Ex - b\| = X(\mu^*) \cap \mathcal{H}_{E,b},$$

therefore, in exact arithmetic, we are able to calculate a primal optimal x^*, as an example with the projected conjugate gradient algorithm, where the box constraints are described by $X(\mu^*)$.
In a more realistic scenario[5], we have a μ_{lb} and a lower bound to the optimal value $L(\mu_{lb}) \leq \nu^*(D1)$; from this we can calculate the corresponding x_{lb}. The simple geometrical insight is that, if we are near to the optimal μ^* and L is not differentiable in μ^*, then a min-norm ϵ-subgradient could correspond to an x' nearer to the optimal primal point than x_{lb}.
In practice, consider the box $X_\epsilon(\mu) \supseteq X(\mu) = X_0(\mu)$, equal to $X(\mu)$ except for the fact the any component j corresponding to $Q_{jj} = 0$ is defined by

$$X_\epsilon(\mu)_j := \begin{cases} \{l_j\} & \text{if } (q + E^\intercal \mu)_j > \epsilon \\ \{u_j\} & \text{if } (q + E^\intercal \mu)_j < -\epsilon \\ [l_j, u_j] & \text{otherwise.} \end{cases}$$

Does a $\Delta\mu$ exist such that $X(\mu + \Delta\mu) = X_\epsilon(\mu)$? Let v be the vector in $\mathbb{R}^n$ defined by

$$v_j := \begin{cases} -(q + E^\intercal \mu)_j & \text{if } Q_{jj} = 0 \text{ and } |(q + E^\intercal \mu)_j| \leq \epsilon \\ 0 & \text{otherwise;} \end{cases}$$

then clearly it should hold $E^\intercal \Delta\mu = v$, which is solvable in $\Delta\mu$ if and only if the sum of the changes described by v in any cycle of the graph is null, that is $q^\intercal v = 0$ for any $q \in \ker E$. If any solution exists, it can be found inside a ball of radius $\sqrt{\binom{m}{2}}\,\|v\| \leq \epsilon\sqrt{\binom{m}{2}\dim\ker Q}$ centered in μ.
Though there's no guarantee that $X_\epsilon(\mu)$ is going to describe the subdifferential of the Lagrangian dual in a nearby point, the key observation is that redistributing flow through the network with capacity constraints restricted to $X_\epsilon(\mu)$, the Lagrangian variation is bounded by $\epsilon \left\|u_{X_\epsilon(\mu)} - l_{X_\epsilon(\mu)}\right\|_1$; the smaller the variation in the Lagrangian, the more we are moving toward the saddle point. Therefore the strategy is similar to the first order min-cost heuristic, where the costs are given by the gradient of the Lagrangian, but redistribution of flow along arcs without a quadratic part is prioritized. The nearer the starting point to the optimal solution, the more the strategy is expected to be effective.

3. Subgradient Iterations

Classical subgradient iterations are the baseline upon which to build deflected iterations and against which to compare their performance. We deal with concave maximization problems, so we describe subgradient methods as applied to concave functions; for coherence, we will stick to the notation introduced for the problem $(D1)$ and denote the concave function under examination with L, the variable with μ, with $g(\mu) \in \partial L(\mu)$ any subgradient in the subdifferential $\partial L(\mu)$.

Convergence: general results ▶ Subgradient methods are very simple iterations usually written in one of the forms:

$$\mu^{k+1} = \mu^k + \alpha_{k+1} g(\mu^k) \qquad \text{(StepSize)},$$

$$\mu^{k+1} = \mu^k + \gamma_{k+1} \frac{g(\mu^k)}{\|g(\mu^k)\|} \qquad \text{(StepLength)},$$

where $\alpha_k, \gamma_k > 0$. Since subgradient methods are not strictly descent methods, we have to keep track of the best result found so far, i.e. the one with greatest function value, as of

[5] A full-fledged scenario would include floating point errors, e.g. a lower bound is such only up to a given approximation, to be evaluated

$$\mu_{best}^k = \arg\max(L(\mu^1), \cdots, L(\mu^k)),$$
$$L_{best}^k = L(\mu_{best}^k) = \max(L(\mu^1), \cdots, L(\mu^k)).$$

Let L^* be the optimal value; from the definition, it's simple to show [7] that

$$L^* - L_{best}^k \leq \frac{\left\|\mu^1 - \mu^*\right\|^2 + \sum_{i=1}^k \alpha_i^2 \left\|g(\mu^i)\right\|^2}{2\sum_{i=1}^k \alpha_i}.$$

If a bound for the distance of the optimal point, $\left\|\mu^1 - \mu^*\right\| \leq R$, and a bound for the subgradient, $\forall\mu \ \|g(\mu)\| \leq G$, are available, i.e. equation (3), we obtain

$$L^* - L_{best}^k \leq \frac{R^2 + G^2 \sum_{i=1}^k \alpha_i^2}{2\sum_{i=1}^k \alpha_i}. \tag{4}$$

Note that the right-hand side of (4) is a convex function of the α_i, so that it has minimum when $\forall i, \ \alpha_i = \alpha$, that is

$$L^* - L_{best}^k \leq \frac{R^2}{2k\alpha} + \frac{G^2\alpha}{2};$$

fixed an ϵ, we find for $\alpha = \frac{\epsilon}{G^2}$ and $k = \frac{G^2R^2}{\epsilon^2}$ that $L^* - L_{best}^k \leq \epsilon$; fixed the number of iterations k, we have for $\alpha = \frac{R}{G\sqrt{k}}$ that

$$L^* - L_{best}^k \leq \frac{RG}{\sqrt{k}}.$$

The upper bound is then confirming the intuitive idea that the step size should decrease together with the distance from the optimal value.
If we know the optimal value L^*, or an estimate for it, we can make our choice be adaptive with respect to the number of iterations, by maximizing the objective value step by step:

$$\gamma_{k+1} = L^* - L(\mu^k) \implies \sum_{i=1}^k (L^* - L(\mu^i))^2 \leq R^2G^2, \tag{PolyakStep}$$

that is,

$$L^* - L_{best}^k \leq \frac{RG}{\sqrt{k}}.$$

In fact, a theorem by Nesterov shows that, for general concave functions with bounded subgradients, this is optimal up to a constant factor.

Theorem 0.3 Nesterov [4] Thm 3.2.5. *Suppose μ^{k+1} is computed by an* arbitrary *method as*

$$\mu^{k+1} = \mu^1 + span\{g(\mu^1), \cdots, g(\mu^k)\}$$

where the $g(\mu^i) \in \partial L(\mu^i)$ are arbitrary. Then there is a nonsmooth concave function with $\|g\| \leq G$ uniformly so that the above method obeys

$$L^* - L(\mu^k) \geq \frac{\left\|\mu^1 - \mu^*\right\| G}{2\sqrt{k}}.$$

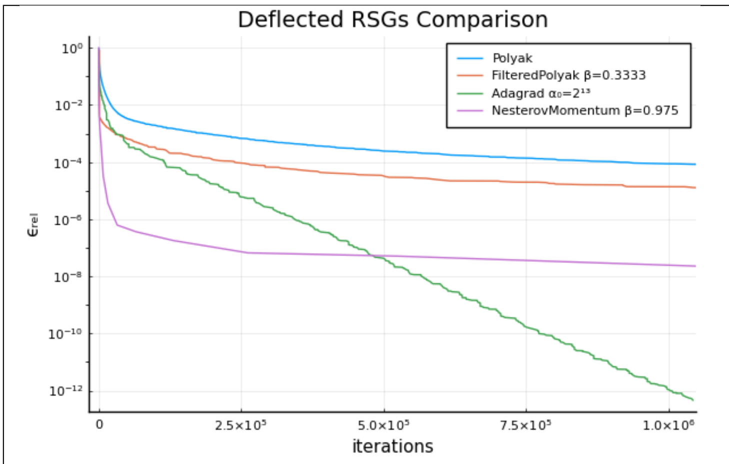
The Polyak iterations reach the optimal worst case for generic concave functions. However, our objective function is not a generic concave function: it is piecewise quadratic/linear. Can we do better? Indeed, if we take a look at how the Polyak approach works on any of our instances, we get robust results, but still incredibly slow - as evident in the first row of Table 1. Similarly, the deflection methods built upon Polyak step sizes did not bring remarkable improvements on the convergence speed. It was while playing with the automatic tuner that a different strategy emerged.

4. Restarted Subgradient Methods

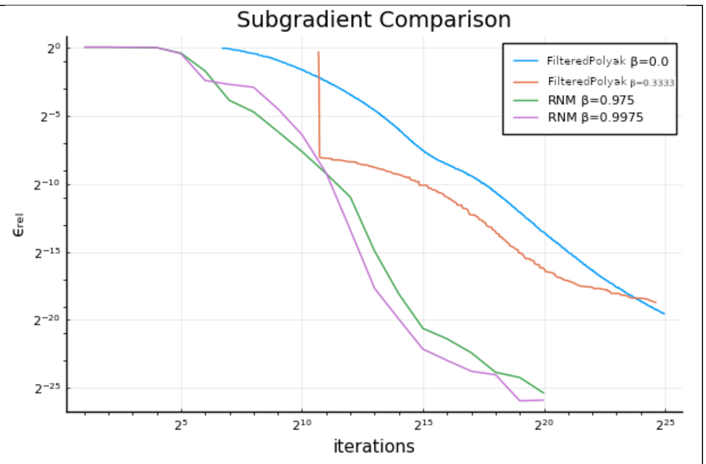
Our investigations on dynamic parameter tuning, i.e. Table 2, in deflected subgradient methods led to the experimental evidence that a strategy based on restarts and exponential decay of the step size yields a robust, efficient and simple enough method. Interestingly, in the case of standard subgradient iterations, with fixed step size, there is a nice analysis of the reasons for such effectiveness.
A restarted subgradient method (RSG) runs in multiple stages warm-started by the solution from previous stages. Within each stage, the subgradient iteration of choice is performed for a fixed number of times with a constant step size. This step size is reduced geometrically from stage to stage, as described in Algorithm 1. Theorem 0.4 links the restarted subgradient complexity to the distance between the ϵ-level set and the optimal set; because of such property, specific iteration complexity are deduced based on the local growth property of the objective function in Corollary 0.5.
Let $\mathcal{L}_\epsilon$ denote the ϵ-level set of L, i.e. $\mathcal{L}_\epsilon := \{\mu : L(\mu) = L^* + \epsilon\}$, and let $\rho_\epsilon := \min_{\mu \in \mathcal{L}_\epsilon} \|\partial L(\mu)\|$. We denote with μ^* the nearest optimal point to μ.

Table 1

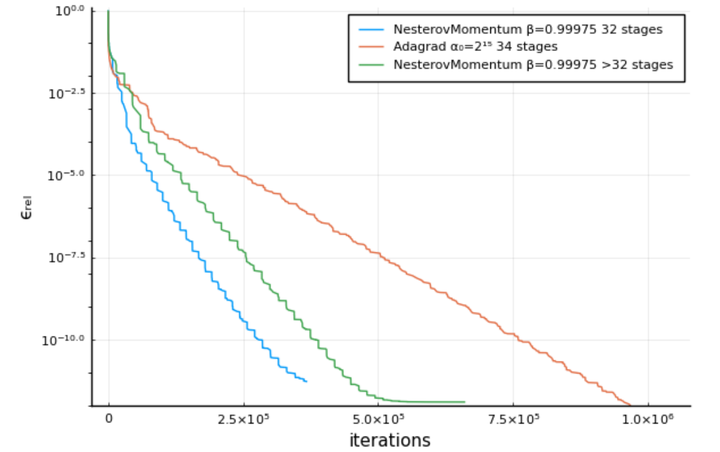
Restarted deflected methods: pictures of an exhibition. With no tuning, RNM is quite robust. The less the instance is singular, the more the method is sensible to parameter tuning. With parameter tuning, RNM is consistently the most efficient method - though other methods may have a smoother dynamics.



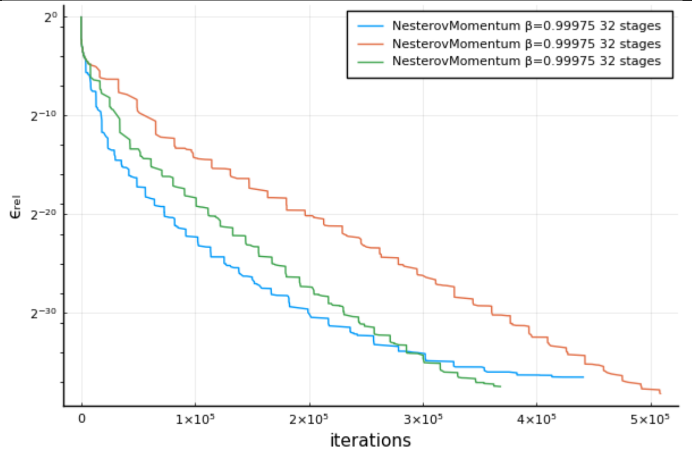
Nontuned RestartedNesterovMomentum vs other best-tuned methods: `sing=0.66`.



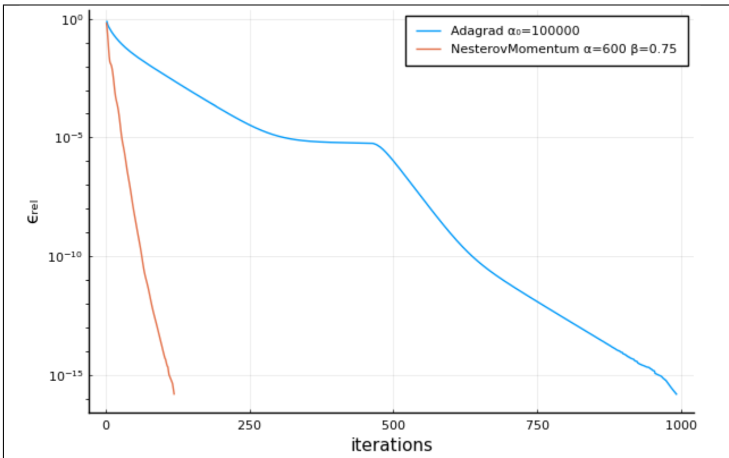
Best-tuned Polyaks vs RestartedNesterovMomentum: `sing=0.66`.



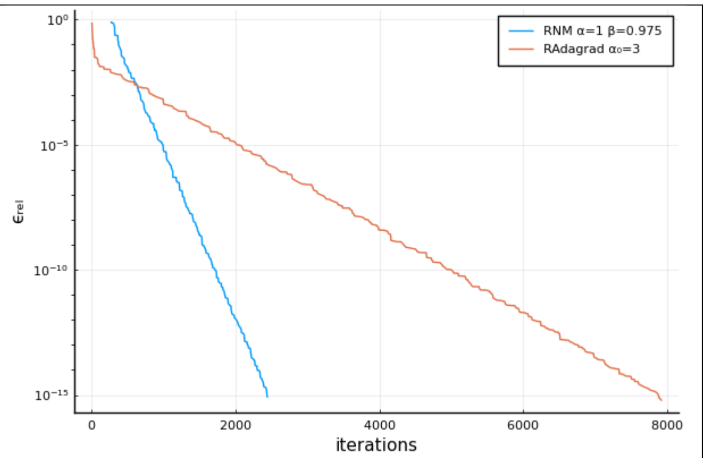
Tuned RestartedNesterovMomentum vs best-tuned RestartedAdagrad: `sing=0.33`. Though Adagrad descends smoothly when well tuned, it is usually possible to make RNM twice as fast.



Different iteration schemes of RestartedNesterovMomentum: usually later stages of RNM require more iterations, as confirmed by faster converging lines for iter $[\text{stage}] = cx^{\text{stage}}$.



NesterovMomentum vs Adagrad: `sing=0.0`. Especially for nonsingular instances, the full power of NesterovMomentum is unleashed by tuning also the α parameter. Here compared to Adagrad best-tuned.



RestartedNesterovMomentum vs best-tuned RestartedAdagrad: `sing=1.0`. For singular instances, RNM is quite robust to parameters; we never gained in convergence speed by tuning α.

Comparison of deflected restarted methods.

Theorem 0.4 RSG [12] Thm 3. *Under the assumptions*

- $\forall \mu \; \|g(\mu)\| \leq G$,

- *the subgradient iterations will stay inside a region Ω such that $\forall \mu \in \Omega$ we have $L^* - L(\mu) \le \epsilon_0$,*
- *the optimal set is non-empty convex compact*[6],

the total number of iterations for Algorithm 1 to find a 2ϵ-optimal solution is at most $O\left(t \left\lceil \log_r \left(\frac{\epsilon_0}{\epsilon}\right)\right\rceil\right)$, where $t \ge \frac{r^2 G^2}{\rho_\epsilon^2}$. Denoting with $B_\epsilon := \max_{\mu \in \mathcal{L}_\epsilon} \|\mu - \mu^*\|$, we have that $\rho_\epsilon \ge \frac{\epsilon}{B_\epsilon}$, so that the iteration complexity for obtaining a 2ϵ-optimal solution is $O\left(\frac{r^2 G^2 B_\epsilon^2}{\epsilon^2} \left\lceil \log_r \left(\frac{\epsilon_0}{\epsilon}\right)\right\rceil\right)$.

Algorithm 1: Restarted Subgradient

Input: the number of stages K and the number of iterations t per-stage, μ_0 and $r > 1$

Set $\alpha_1 = \frac{\epsilon_0}{rG^2}$ // α : step size

for $k = 1, \dots, K$ **do**
  Call subgradient subroutine SG to obtain $\mu_k = SG(\mu_{k-1}, \alpha_k, t)$
  Set $\alpha_{k+1} = \frac{\alpha_k}{r}$
end
Output: μ_K

In practice, an estimate of all the parameters in theorem 0.4 is not needed, i.e. an exponential search is enough to determine the number of iterations per stage required for a desired ϵ.
A nice corollary follows from theorem 0.4.

Corollary 0.5 RSG with Local Error Bounds [12] Crl 7. *If, for some constants $\theta \in (0, 1]$ and $c > 0$,*

$$\|\mu - \mu^*\| \le c(L^* - L(\mu))^\theta, \quad \forall \mu : L^* - L(\mu) \le \epsilon,$$

then the iteration complexity for the RSG algorithm of Theorem 0.4 for obtaining a 2ϵ-optimal solution is $O\left(\frac{r^2 G^2 c^2}{\epsilon^{2(1-\theta)}} \log_r \left(\frac{\epsilon_0}{\epsilon}\right)\right)$, provided $t = \frac{r^2 G^2 c^2}{\epsilon^{2(1-\theta)}}$ and $K = \left\lceil \log_r \left(\frac{\epsilon_0}{\epsilon}\right)\right\rceil$.

Given that the local growth of $\mu - \mu^*$ as a power of $L(\mu) - L^*$ has an exponent $\theta \in \left[\frac{1}{2}, 1\right]$, we would expect for the number of iterations t per stage to satisfy approximately $t \propto \epsilon^{-\vartheta}$, where $\vartheta \in [0, 1]$ depends on $\dim \ker Q$.

5. Deflected Subgradient Iterations

Restarted subgradient methods turned out to be both simple to tune (enough) and effective approaches. They allowed to exploit the particular geometry of the problem at hand. In order to further accelerate subgradient methods we try to contrast the common zig-zagging behaviour by playing with the direction the algorithm chooses at each iteration. In the literature such methods are known as deflected subgradient methods. They incorporate some form of memory (and of premonition, in the case of NesterovMomentum) of the previous iterations to synthesise a hopefully more promising direction for the step. We describe a couple of alternatives which approach the problem from different points of view; the description of each iteration is in the following table.

Name	Params	Iteration
Adagrad	$\alpha \in [0, 1]$	$g^{k+1} \in \partial L(\mu^k)$ $s^{k+1} \leftarrow s^k + g^{k+1}.* g^{k+1}$ $d^{k+1} \leftarrow g^{k+1}./.\sqrt{s^{k+1}}$ $\mu^{k+1} \leftarrow \mu^k + \alpha d^{k+1}$
RMSProp	$\alpha \in [0, 1]$ $\gamma \in [0, 1]$	$g^{k+1} \in \partial L(\mu^k)$ $s^{k+1} \leftarrow \gamma s^k + (1-\gamma) g^{k+1}.* g^{k+1}$ $d^{k+1} \leftarrow g^{k+1}./.\sqrt{s^{k+1}}$ $\mu^{k+1} \leftarrow \mu^k + \alpha d^{k+1}$
Nesterov Momentum	$\alpha \in [0, 1]$ $\beta \in [0, 1]$	$g^{k+1} \in \partial L(\mu^k + \beta v^k)$ $v^{k+1} \leftarrow \beta v^k + \alpha g^{k+1}$ $\mu^{k+1} \leftarrow \mu^k + v^{k+1}$

[6]In our case, the optimal set is affine, in particular not compact. However, each direction generating the optimal set is orthogonal to the whole $\partial L(\mu)$ everywhere. Hence the algorithm is not affected by the degeneracy of the optimal set.

Both Adagrad and RMSProp iterations reduce the movement along coordinates which are moving more than the others, see [13] for the theoretical justification of Adagrad, while RMSProp has been devised to obviate to the dampening of Adagrad. Incorporating them in a restarting scheme with decay of the step size parameter led to nice results. One of the most performing iterations appeared to be the NesterovMomentum with high momentum, for which we could not retrieve a proof of convergence. Hence we present our short tentative convergence study, and *leave as an unsolved question the theoretical convergence of the method.*

NesterovMomentum: tentative analysis ▶ At iteration k, the step direction is the sum of a momentum term, βv^k, and the contribution αg^{k+1} from the look-ahead subgradient, $g^{k+1} \in \partial L\left(\mu^k + \beta v^k\right)$. Unrolling the iteration equations, we derive

$$v^k = \alpha \sum_{i=1}^{k} \beta^{k-i} g^i,$$
$$\mu^k = \mu^0 + \sum_{i=1}^{k} v^i = \mu^0 + \alpha \sum_{i=1}^{k} g^i \frac{1-\beta^{k+1-i}}{1-\beta},$$
$$g^k \in \partial L\left(\mu^0 + \alpha \sum_{i=1}^{k-1} g^i \frac{1-\beta^{k+1-i}}{1-\beta}\right).$$

Let's analyse if a reduction to the convergence results for ϵ-subgradients is viable. By the information transport property of ϵ-subgradients,

$$g \in \partial_\epsilon L(\mu) \implies g \in \partial_{\epsilon'} L(\mu'),$$
$$\epsilon' = \epsilon + g \cdot (\mu' - \mu) - L(\mu') + L(\mu),$$

for the look-ahead subgradient we can write

$$g^{k+1} \in \partial_{\rho^{k+1}} L\left(\mu^k\right),$$
$$\rho^{k+1} = L(\mu^k + \beta v^k) - L(\mu^k) - \beta g^{k+1} \cdot v^k,$$

while, for the momentum term, we obtain the formula

$$\frac{1-\beta}{\alpha(1-\beta^k)} v^k \in \partial_{\sigma^k} L\left(\mu^{k-1}\right) \implies \frac{1-\beta}{\alpha(1-\beta^k)} v^k \in \partial_{\sigma^{k'}} L\left(\mu^k\right),$$
$$\sigma^{k'} = \sigma^k + L\left(\mu^{k-1}\right) - L\left(\mu^k\right) + \frac{1-\beta}{\alpha(1-\beta^k)} v^k \cdot v^k.$$

Finally, from $\frac{v^{k+1}}{\alpha(1+\cdots+\beta^k)} = \frac{1}{\alpha(1+\cdots+\beta^k)}\left(\alpha g^{k+1} + \beta\alpha(1+\cdots+\beta^{k-1})\frac{v^k}{\alpha(1+\cdots+\beta^{k-1})}\right)$, we deduce

$$\frac{v^{k+1}}{\alpha(1+\cdots+\beta^k)} \in \partial_{\sigma^{k+1}} L\left(\mu^k\right),$$
$$\sigma^{k+1} \le \max(\rho^{k+1}, \sigma^{k'}) \le \rho^{k+1} + \sigma^{k'} \implies$$
$$\implies \sigma^{k+1} \le \rho^{k+1} + L(\mu^{k-1}) - L(\mu^k) + \frac{1-\beta}{\alpha(1-\beta^k)} v^k \cdot v^k + \sigma^k \implies$$
$$\implies \sigma^{k+1} \le L(\mu^0) - L(\mu^k) + \sum_{i=1}^{k}\left(\rho^{i+1} + \frac{1-\beta}{\alpha(1-\beta^i)} v^i \cdot v^i\right).$$

It is not clear at this point how/if it is possible to reduce the proof of convergence of RestartedNesterovMomentum to the proof of convergence of Restarted(Approximate)Subgradients.

Arithmetic cost per iteration ▶ In the next table we describe the arithmetic cost of the main iterations we discussed; this allows to estimate, from the number of needed iterations, the performance of the algorithm. Exploiting the next table we can evaluate how each subgradient method is performing from a simple iteration count, with no need of a optimized implementation.

Iteration	$+$	$*$	max / min	$/$	$\sqrt{\cdot}$
Subgradient	$5n+2m$	$n+m$	$2n$	0	0
NesterovMomentum	$5n+4m$	$n+3m$	$2n$	0	0
Adagrad	$5n+3m$	$n+2m$	$2n$	m	m
RMSProp	$5n+3m$	$n+4m$	$2n$	m	m

6. Experiments

Implementation ▷ The software is distributed as a Julia package. It is designed to allow for an easy exploration of the dynamics of (deflected) subgradient iterations, heuristics and meta-algorithms on this particular problem; because of this, the resulting product is not a 1-click solver and the focus is on flexibility rather than performance. A simple parallelized C++ implementation of the Restarted Nesterov Momentum is accompanying the Julia package; in our setup the parallel primitives of the C++ STL are leveraging Intel TBB libraries, with no gpu support at the time of writing. This allowed to test the algorithm on larger instances.

Testing material ▷ Separable convex quadratic min cost flow instance generators are freely available on the web, together with some already generated test-sets[7]. In addition to the available material, the simple random test generator used in our experiments is described in Algorithm 2; note that it generates a possibly disconnected multigraph.

Algorithm 2: Sketch of a Test Generator for QMCFBProblem

`/* m : number of nodes */`
`/* n : number of arcs */`
`/*` $\mathfrak{s}$ `:` $\ker Q$ `*/`
`/*` $\mathfrak{a}$ `:` $\approx$ `active box constraints */`
function `get_test`(m, n, $\mathfrak{s}$, $\mathfrak{a}$):
  E = incidence matrix $m \times n$ with random arcs
  x = random flow
  (l, u) = random capacity interval around x
  b = E*x `// so x is an internal feasible point`

  Q[$\bar{\mathfrak{s}}$, $\bar{\mathfrak{s}}$] = random diagonal $\succ 0$
  Q[$\mathfrak{s}$, $\mathfrak{s}$] = 0
  ($\mathfrak{l}$, $\mathfrak{u}$) = random partition of $\mathfrak{a}$
  q[$\mathfrak{l}$] = random > -Q[$\mathfrak{l}$, $\mathfrak{l}$] * l[$\mathfrak{l}$]
  q[$\mathfrak{u}$] = random < -Q[$\mathfrak{u}$, $\mathfrak{u}$] * u[$\mathfrak{u}$]
  q[$\bar{\mathfrak{a}}$] = random $\in$ (-Q[$\bar{\mathfrak{a}}$, $\bar{\mathfrak{a}}$] * u[$\bar{\mathfrak{a}}$], -Q[$\bar{\mathfrak{a}}$, $\bar{\mathfrak{a}}$] * l[$\bar{\mathfrak{a}}$])

  return (Q, q, E, b, l, u)

Errors ▷ Calculations are carried out in the relaxed space of $(D1)$; points are exactly inside the box, but aren't usually exactly in $\mathcal{H}_{E,b}$, even after projection. To estimate the error on the objective value upper bound, for a point $x \in \mathcal{H}_{E,b+\Delta b}$, we use the estimate for $d(x, S_{(P)})$ from proposition 0.2 together with $|f(x+\Delta x) - f(x)| \approx |\partial f(x)\Delta x| \leq \|Qx + q\| \, \|\Delta x\|$.

Discarded strategies ▶ Most of the methods implemented in the package did not reveal to be effective for the problem at hand. In particular, we stress upon the following ones:

- *Projected conjugate subgradient* ▷ Since the Lagrangian dual is piecewise-quadratic, it's possible to define an exact line search; moreover, in exact arithmetic, once the right polyhedral region is found, with the conjugate gradient it could be possible to find the optimal point. The resulting conjugate gradient algorithm works fine when $Q \succ 0$, details can be found in the literature [11].
- *Ergodic sequences* ▷ There are general methods to derive primal iterates from the dual ones [3] [5] [6]; in particular we had implemented and tested the iterations described as "harmonic sequences" in [5]. However, the techniques described in §2 appeared to be superior as for quality (stricter upper bounds to the optimal value) and flexibility (completely independent from the subgradient iteration of choice).
- *Meta algorithms* ▷ Deflected subgradient methods are empirically sensitive not only to parameters, but, by construction, also to the history of the iterations. The iterations can be embedded in a meta-algorithm which drives the dynamics of the parameters and the restarts (where the memory of the iteration is cleared). We tested the idea of fine-tuning with a zero order parameter search, the Nelder-Mead simplex algorithm. For some deflected methods, investigations suggested the very simple, but quite performant, restart and parameter dynamics described in the foregoing sections, hence supplanting the idea of actually using the meta-algorithms to drive the subgradient method parameters.
- *Stopping criteria* ▷ Projection on the feasible polyhedron §2, as implemented at the time of writing, is computationally too expensive to be used as a stopping criterion. At the same time, the standard upper bounds for subgradient iterations are too loose to be useful. A possible strategy to attain efficient error-dependent projections could be to implement a scaling linear min-cost flow algorithm, like the Goldberg-Tarjan one, where the computational complexity is dependent on the desired precision, to be set accordingly to the bounds in Proposition 0.2.

[7] As an example, the data available at the MCF web page by the Operations Research Group of University of Pisa.

Selected data ▶ About notation: we characterize each problem instance generated by Algorithm 2 with the following attributes:

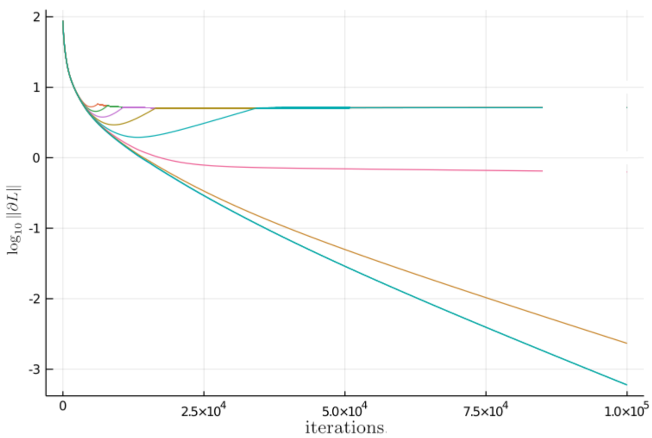
- `nodes`: number of nodes
- `arcs`: number of arcs
- `singular`: (fraction of) zero elements on the diagonal of Q
- `active`: expected (fraction of) active constraints at the optimal point.

Instances generated with Pargen+Netgen+Qfcgen have a standardized name of the form `netgen`-n-ρ-k-cf-cq-$scale$-$singular$, where:

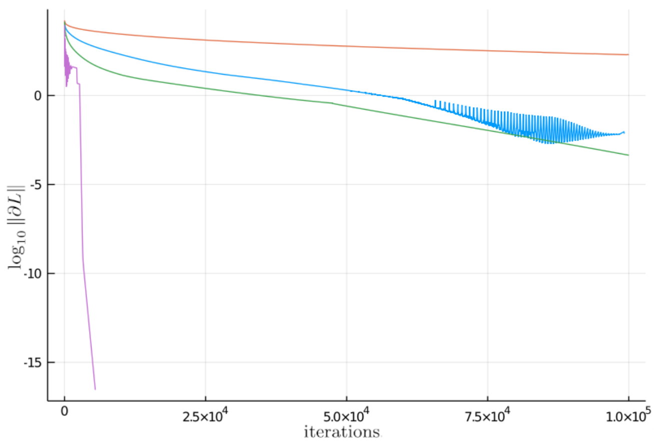
- ρ: then we have for the number of nodes $m = \left\lfloor \frac{1+\sqrt{1+\frac{32n}{\rho}}}{2} \right\rfloor$
- k: instance number
- $cf \in$ `[a, b]`: indicates if the fixed costs are generated to be high (`a`) or low (`b`) with respect to the linear costs generated by netgen
- $cq \in$ `[a, b]`: indicates if the quadratic costs are generated to be high (`a`) or low (`b`) with respect to the fixed costs
- $scale \in$ `[s, ns]`: if `s`, capacity are scaled by 0.7
- $singular$: number of diagonal elements of Q set to 0

We do not consider fixed costs in the present work, thus cf and cq together regulate the ratio between Q and q, in particular `a-a` corresponds to the highest ratio $\frac{Q}{q}$, `b-b` to the lowest one.

Deflected Subgradient methods ▷ A striking property of standard, non restarted, deflected subgradient methods, is their sensitivity to the parameters.
This appears to be even more evident when the dual problem is C^1 (we hint at [14] as a possible way to investigate the particular dynamics), hence we corroborate the statement with two anecdotal examples where Q is nonsingular. The precision attained for the solution is here measured with $\|\nabla_\mu L(\mu)\|$, which is more strictly related to the dynamics described by the iterations than the distance from the optimal point.



Dynamics of Nesterov Momentum iterations while varying the parameters. Each line represents the dynamics for a different parameter setup. The best configuration we localized appeared to be a stationary point in parameter space.



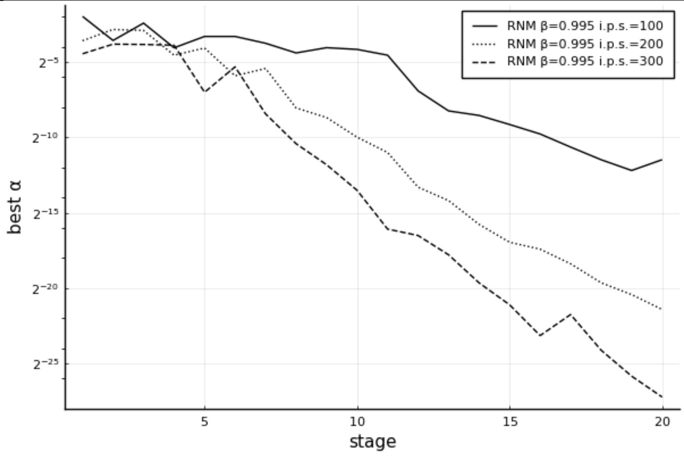
Dynamics for the RMSProp iterations. By driving the parameters a bit further the stabler stationary position, we stumbled very near to the optimal point very soon.

After an initial investigation, with the help of the automatic parameter tuner, over all the implemented deflected subgradient methods (which can be found in `subgradient.jl`), we realized that the most performing easy-tunable deflected method was consistently the NesterovMomentum iteration, with sufficiently high momentum parameter, i.e. $\beta \geq 0.95$.

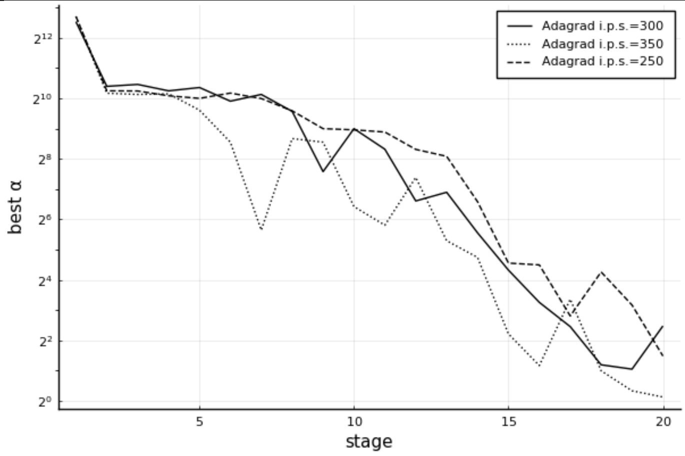
Toward restarted subgradients ▷ We have seen that encapsulating subgradient iterations in a exponentially decaying stepsize scheme brings a new shape-dependent flavour to the classical, fixed stepsize, subgradient method. In practice, notwithstanding the lack of a proof of convergence, the NesterovMomentum was easy to adapt to an analogous restarting scheme: the exponential decay of the stepsize parameter α is accompanied by the reset of the iteration memory ($v \leftarrow 0$), leaving the momentum parameter β constant through the stages.

Table 2

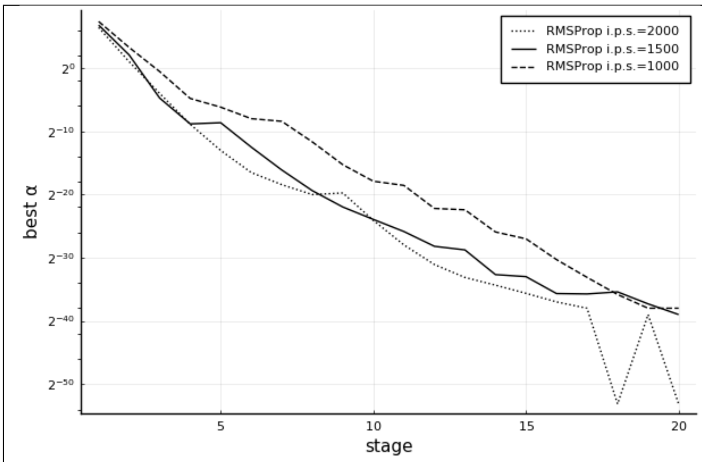
Restarted methods: per-stage parameter tuning, fixed number of iterations per stage. Unpolished results from the NelderMead searcher, `i.p.s`=iterations per stage, instance: `netgen-1000-1-1-b-a-ns-0660`.

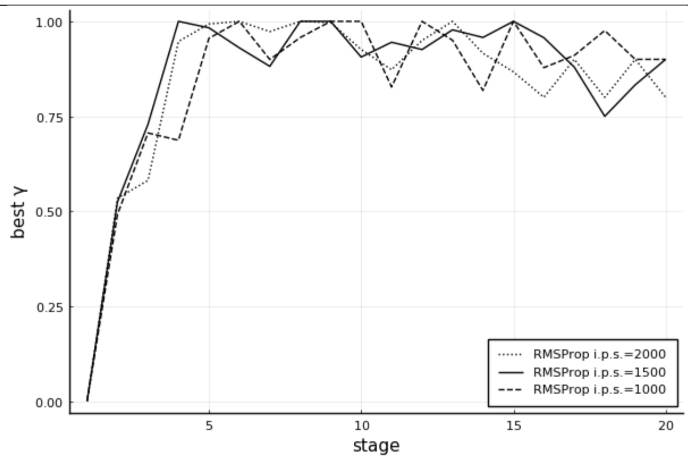


NesterovMomentum, fixed β.

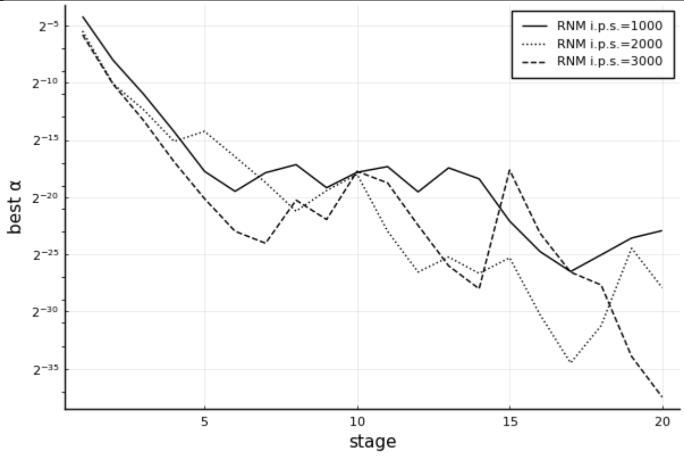


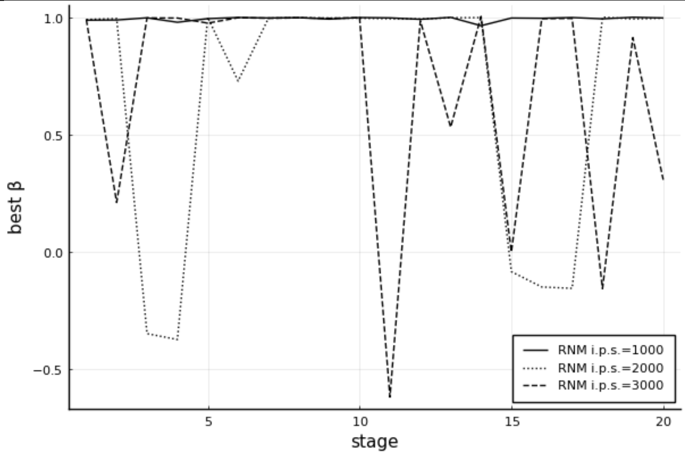
Adagrad. The dampening of Adagrad leads to $\alpha \notin [0,1]$ for all the best configurations.





RMSProp.





NesterovMomentum.

Let's take for granted the summentioned skeleton: we fix a number of stages, each of which runs the same number of iterations; at the beginning of each stage the memory of the deflected subgradient is reset (we do not provide here evidence supporting the memory reset, but this is easily accomplished with the package). With the automatic tuner, we can analyse the dynamics of the optimal parameters from a stage to the next one; aware of the fact that the tuner is driving the parameters toward *locally* optimal values, we can read the resulting plots as a hint to the decay of the step-size parameter α - decay compatible with an exponential one - but also to some ranges for the remaining parameters. Table 2 shows the unpolished results for NesterovMomentum, Adagrad and RMSProp; to render the general trend more efficiently, in each figure three lines corresponding to a different choice of the number of iterations per stage (`i.p.s`) are plotted. The rules that we may extrapolate include the exponential decay for the stepsize parameter (α), a constant value for β very near to 1 for the NesterovMomentum and similarly, but less clearly, a value for γ

in RMSProp compatible with being constant and greater than 0.75 - the suggestion found in literature. It should be remarked that, because of the dampening effect of the algorithm, Adagrad shows an optimal stepsize parameter that, although exponentially decreasing, is greater than 1 - we did not explore how the optimal starting value for α is correlated to the macroscopic characteristics of the instance at hand. Lastly, the optimal ratio for the exponential decay is dependent on the number of iterations per stage. When not otherwise specified, *subgradient methods are run with standard parameter setup (not tuned).* A possibly relevant improvement (see Figure 2) could be smart cheap tuner.

For methods with more than 2 parameters to be tuned, it is harder to obtain general prescriptions leading to good performance; nonetheless the results are mostly compatible with an exponential decay of the stepsize parameter.

Intuitively, the decreasing of the stepsize parameter is useful if we are near enough to the optimal set; to confirm the idea, we can run repeatedly the restarted algorithm for a prescribed number of stages, each time with an greater number of iterations per stage. We expect later stages to be useful to approach the optimal set only if the iterations per stage are enough. In Figure 1 we realize the plan by running the restarted subgradient on the set $2^{[0:15]}$ of iterations per stage. Results are compatible with Corollary 0.5:

- It is necessary to increment the number of iterations per stage to get nearer to the optimum - in practice, later stages require more iterations. This is qualitatively compatible with the formula $t = \frac{r^2 G^2 c^2}{\epsilon^{2(1-\theta)}}$.
- The attainable precision is exponential in the number of stages, where the base is given by the geometric reduction factor of the step-size parameter. The plot confirms the worst-case formula $K = \left\lceil \log_r \left(\frac{\epsilon_0}{\epsilon}\right) \right\rceil$.

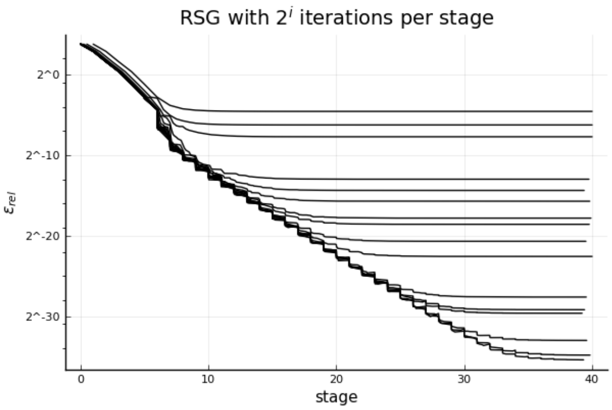


FIGURE 1. Each line represents the relative error for the RestartedSubgradient with fixed number of iterations per stage, $t = 2^i$, $i \in \{0, 1, \cdots, 15\}$, over the 40 stages. This supports the idea that t should increase as an inverse power of ϵ_{rel} and that the number of (useful) stages is approximately $\log_r \left(\frac{\epsilon_0}{\epsilon}\right)$. Instance: `singular`=0.55, `active`=0.87.

The plot also suggests to try the restarted approach with a number of iterations per stage that is exponentially increasing from a stage to the next one rather than fixed; this is still to be explored, a minor confirmation can be found in Table 1. It could lead to a speedup proportional to the number of stages.

We tried to measure how much the relation between the number of iterations per stage t and the desired error ϵ depends on the geometry of the problem; this should be evident from the plots of the relative error against the iterations per stage for instances of different singularity, but in practice we did not collect empirical evidence of the formula in Corollary 0.5. Instead, what is evident from data is that, since nonsingularity of Q implies differentiability of the Lagrangian Dual, nearness to `singular=0` is crucial for the efficacy of the methods.

Sticking to the method we were able to exploit the best, RNM, we can analyse in more details what is the optimal number of stages/iterations per stage. We carry out the analysis on the test-set `netgen1000`, a test-set of `QMCFBProblems` with 1000 arcs, composed of 5 instances per parameter setup, generated with Netgen+Pargen+Qfcgen[8]. In Table 5 the optimal number of stages is plotted against the desired relative error, while in Table 6 the same is shown for the optimal number of iterations per stage. This is accomplished by locating with an exponential search the optimal setup of stages/iterations per stage. The aggregate figures 2 and 3 remark the most evident properties:

- The optimal number of stages is increasing with the singularity of the instance. In the extremal case of a differentiable Lagrangian Dual, a single stage is almost enough, because the gradient norm is decreasing to 0.
- The smaller are the ratio $\frac{Q}{q}$ and the smaller is Q with respect to the other parameters (the farther is the optimal point of the unconstrained primal problem from the feasible space) the higher is the number of stages; vice versa for the number of iterations per stage.

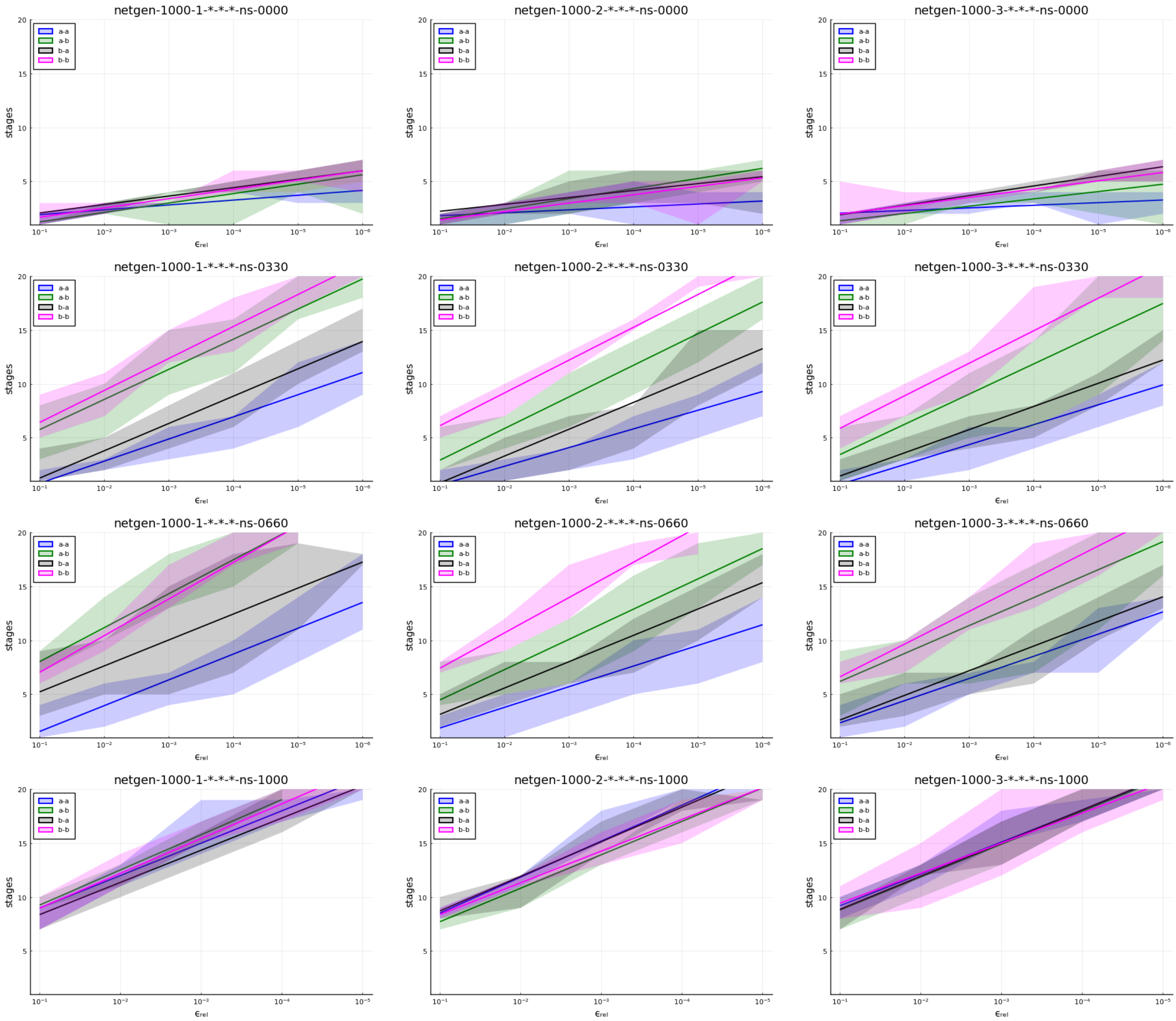
For a better analysis of the summentioned property a dedicated instance generator is needed.

The power of deflection: RNM vs RSG ▷ In Table 3 we compare the performance of RestartedNesterovMomentum versus the one of RestartedSubgradient, on the test-set `netgen1000`. The performance is measured in terms of the number of iterations required to attain a *relative primal-dual gap*, because this is the real precision measure that would be available in practice if we were to exploit the methods we are proposing to solve instances where other solvers fail.

Comparison with an external solver: Gurobi ▷ We tested the performance of the Julia implementation of RestartedNesterovMomentum (RNM) against the commercial solver Gurobi, *specialised for quadratic programs*. Experiments were performed with Julia 1.4.1 running on a Lenovo T430 / 2351AA6 with i5-3320M CPU @ 2.60GHz and 4GiB

[8]The three pass random generator available at the MCF web page

FIGURE 2. ▶ **RNM: optimal stages to reach a given ϵ_{rel}.**



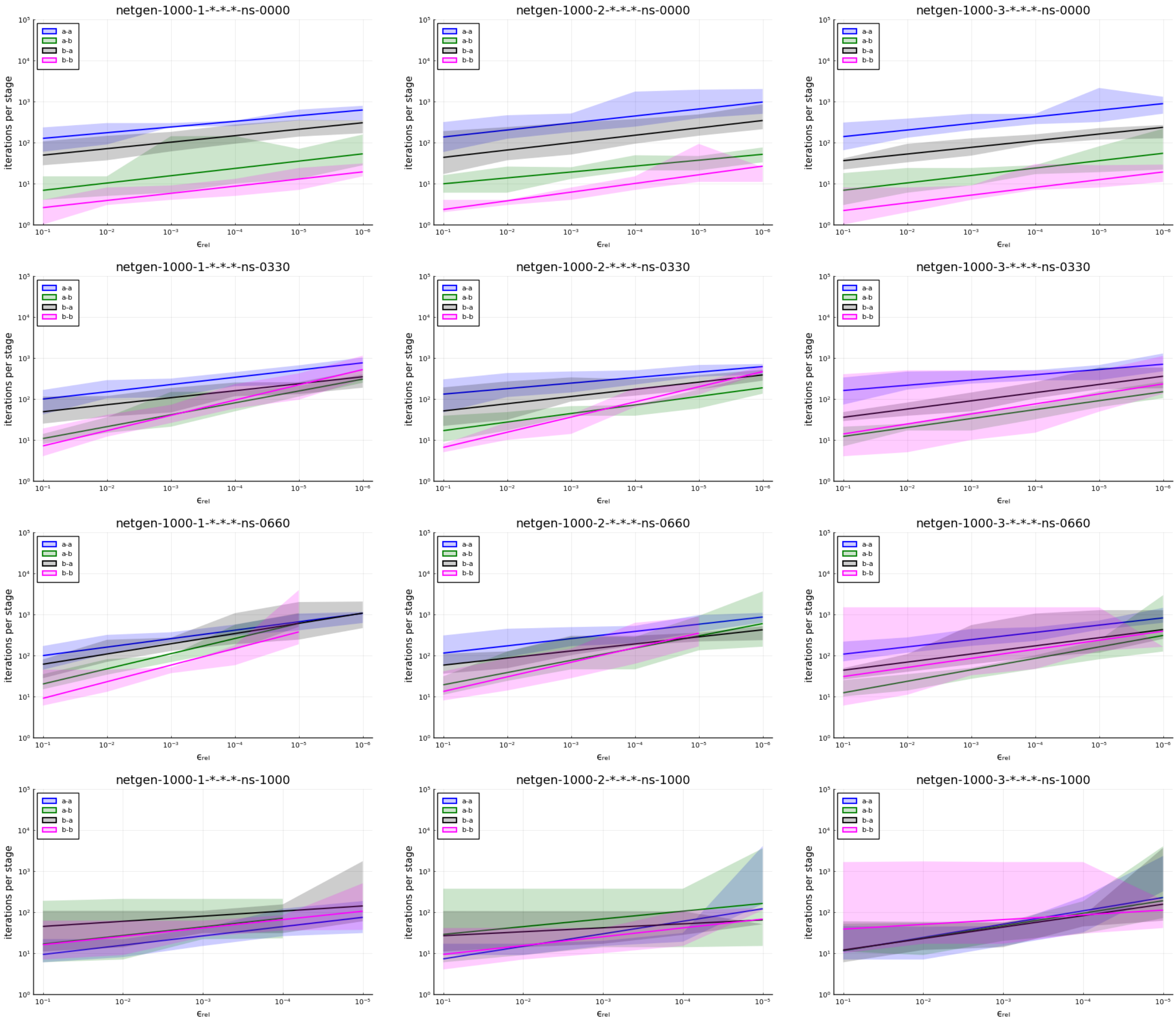
ram. Setting the desired relative precision of RNM to $\epsilon_{rel} < 10^{-8}$, on the test-set `netgen1000`, RNM resulted 10 to 50000 times slower than Gurobi, while for $\epsilon_{rel} < 10^{-5}$, Table 4, it was 7 to 500 times slower on average. Note that the number of stages was fixed to 30, so we could expect that the actual time cost, for $\epsilon_{rel} < 10^{-5}$, could be around half the one we report.

It is interesting to analyse how RNM scales with respect to Gurobi; a glimpse of the scaling behaviour can be deduced from Table 4, comparing the last row to the previous ones. It is evident that, *as of the performance of the package implementation, RNM cannot be considered as a valid alternative to Gurobi even to 5 digits of precision.*

Consequently, to further analyse the scaling behaviour of RestartedNesterovMomentum, we have realized a simple implementation of the algorithm exploiting the parallel directives of C++ STL, run on a workstation assembled with used components from industry, Lenovo P700 / 1030 with E5-2580v3 CPU @ 2.50GHz and 128GiB ram, and built with GCC 9.3.0 on Ubuntu 20.04. We generated another test-set with Netgen+Pargen+Qfcgen composed of instances with number of arcs in $2^{10:17}$; for each number of arcs there are twelve instances described by the combinations of singularity percentage $([0, \frac{1}{3}, \frac{2}{3}, 1])$ and sparsity of the graph (`[a, b, c]`). Results of the experiment are shown in Figure 4; the overall, not surprising, result is that for general instances the method can be considered a valid alternative only up to the first couple of digits, while for non-singular instances it scales better than Gurobi[9] and could be the best option among the two when the number of arcs approaches (and exceeds) 10^5. Since the memory footprint of RNM method is minimal, it could be that for even bigger instances RNM could be the clear winner over Gurobi. We leave this as homework to the interested reader.

[9]We have not analysed the fixed costs from the parallelization, which could artificially enhance the nice-scaling effect. For the smallest instances in the analysis, the single core Julia implementation is even faster than the multithreading ones.

FIGURE 3. ► **RNM: optimal iterations per stage to reach a given ϵ_{rel}.**



For nonsingular instances the Lagrangian dual is C^1, hence all gradient methods could be applied and should be compared to subgradient methods. The geometric intuition is that differentiability improves the convergence of NesterovMomentum - making almost unnecessary the exponential decrease of the step size parameter, because the gradient is converging to 0 as we approach the optimal point. Yet the proposed method works nicely also for "mildly singular" instances: we analyse the convergence of the method for such instances in Figure 5, confirming that, *up to eight digits of precision, RNM may outperform Gurobi on mildly singular instances with more than one million of arcs.*

Table 3

Iterations to $\epsilon_{rel}^{gap} < 10^{-6}$. Test-set generated with Pargen+Netgen+Qfcgen, 1000 arcs, 5 instances per setup.

ρ	cf-cq	sing=0		sing=0.33		sing=0.66		sing=1	
		RSG	RNM	RSG	RNM	RSG	RNM	RSG	RNM
1	a-a	94968±54653	2371±161	737469±36873	15883±2558	888294±44415	148834±96942	46820±40570	9370±7923
2		142372±28263	2148±124	656398±65714	12754±2552	691425±36036	65314±58140	29730±24066	2970±1483
3		97873±24794	2493±447	589860±29493	11278±2061	786634±65731	18240±4916	27341±19887	7725±6252
1	b-a	35039±26218	1739±293	340674±243933	16518±10369	622770±256768	114400±79570	30042±24654	3847±2308
2		13925±7237	1694±339	163872±49147	8847±2190	454868±241871	11281±4102	9888±1903	2290±786
3		22159±18457	1488±182	116120±69950	9574±6240	246111±164086	14245±8469	17092±10995	3001±1591
1	a-b	691±566	361±222	517818±378464	22395±12257	1146590±229685	291165±118518	61930±59037	5901±4375
2		1679±1065	407±133	155124±101844	13052±5401	215410±96123	27387±17658	17612±16225	5573±4174
3		1421±1282	254±89	80969±39950	4366±770	208962±159807	17215±6392	26595±20625	4115±2286
1	b-b	186±65	160±25	144560±29086	34417±27232	531289±351062	83620±58017	40898±35502	5169±3752
2		142±95	153±20	277066±129465	13838±3594	395451±223366	110432±94674	24463±18565	3345±1808
3		168±123	151±44	70481±33609	19925±15124	222825±153010	38057±28566	66447±60029	17621±15941

Table 4

RNM to $\epsilon_{rel} < 10^{-5}$ Vs Gurobi: time in ms. Here we use RNM as implemented in the Julia package and the quadratric solver of Gurobi. Test-set generated with Pargen+Netgen+Qfcgen, 5 instances per setup.

arcs	ρ	cf-cq	sing=0		sing=0.33		sing=0.66		sing=1	
			RNM	Gurobi	RNM	Gurobi	RNM	Gurobi	RNM	Gurobi
1000	1	a-a	3546^{+1230}_{-1362}	31^{+2}_{-2}	3747^{+789}_{-1795}	89^{+161}_{-59}	5381^{+3053}_{-1291}	35^{+6}_{-5}	914^{+1091}_{-631}	30^{+9}_{-8}
	2		4073^{+3385}_{-2158}	43^{+32}_{-15}	3677^{+2925}_{-1849}	34^{+5}_{-4}	6224^{+4085}_{-4261}	34^{+3}_{-2}	570^{+488}_{-277}	26^{+5}_{-2}
	3		3978^{+3715}_{-1953}	30^{+2}_{-2}	4196^{+2499}_{-2361}	34^{+2}_{-3}	4698^{+536}_{-1015}	37^{+2}_{-2}	1348^{+2757}_{-1053}	30^{+9}_{-6}
	1	b-a	1741^{+526}_{-756}	40^{+9}_{-7}	1641^{+704}_{-705}	37^{+3}_{-4}	14099^{+23643}_{-12234}	36^{+9}_{-6}	997^{+1617}_{-755}	26^{+1}_{-1}
	2		1889^{+2438}_{-936}	33^{+5}_{-3}	1580^{+481}_{-537}	35^{+4}_{-4}	2218^{+1496}_{-1159}	45^{+29}_{-17}	511^{+423}_{-251}	28^{+4}_{-3}
	3		1524^{+713}_{-573}	31^{+5}_{-3}	1216^{+872}_{-297}	36^{+5}_{-6}	2509^{+6158}_{-1955}	35^{+3}_{-5}	650^{+413}_{-162}	38^{+46}_{-15}
	1	a-b	302^{+174}_{-62}	35^{+4}_{-4}	1253^{+1116}_{-742}	37^{+5}_{-3}	6693^{+3316}_{-2919}	33^{+3}_{-2}	1539^{+1752}_{-1290}	32^{+31}_{-9}
	2		316^{+63}_{-46}	32^{+6}_{-6}	900^{+491}_{-369}	33^{+4}_{-4}	3628^{+5544}_{-2557}	58^{+51}_{-28}	929^{+1978}_{-689}	30^{+11}_{-5}
	3		289^{+43}_{-52}	35^{+5}_{-3}	572^{+160}_{-105}	31^{+4}_{-2}	2129^{+828}_{-1184}	35^{+4}_{-2}	924^{+998}_{-604}	38^{+26}_{-15}
	1	b-b	284^{+98}_{-48}	33^{+9}_{-5}	2049^{+1494}_{-1077}	43^{+36}_{-15}	2476^{+1878}_{-1479}	35^{+6}_{-4}	911^{+1546}_{-657}	26^{+8}_{-4}
	2		353^{+112}_{-85}	42^{+31}_{-15}	1461^{+686}_{-789}	33^{+3}_{-5}	6345^{+10950}_{-5133}	43^{+28}_{-12}	1160^{+1441}_{-834}	25^{+3}_{-2}
	3		448^{+158}_{-148}	32^{+7}_{-4}	2342^{+7034}_{-2094}	37^{+4}_{-10}	2872^{+5494}_{-2372}	38^{+13}_{-10}	2704^{+6210}_{-2429}	36^{+31}_{-12}
50000	1	a-a	194116	2826	2776811	2867	2843126	4901	2799640	4609

FIGURE 4. ► **Scaling of RNM vs Gurobi.** For singular instances, Gurobi is comparable to RNM run to precision $\epsilon_{rel} < 10^{-2}$. For nonsingular instances RNM scales way better than Gurobi. RNM has also a minimal memory footprint, so we propose its usage for singular instances with more than 10^5 arcs. Figure 5 focus the analysis, confirming the better scaling behaviour of RNM over Gurobi, on a neighbourhood of `sing = 0`.

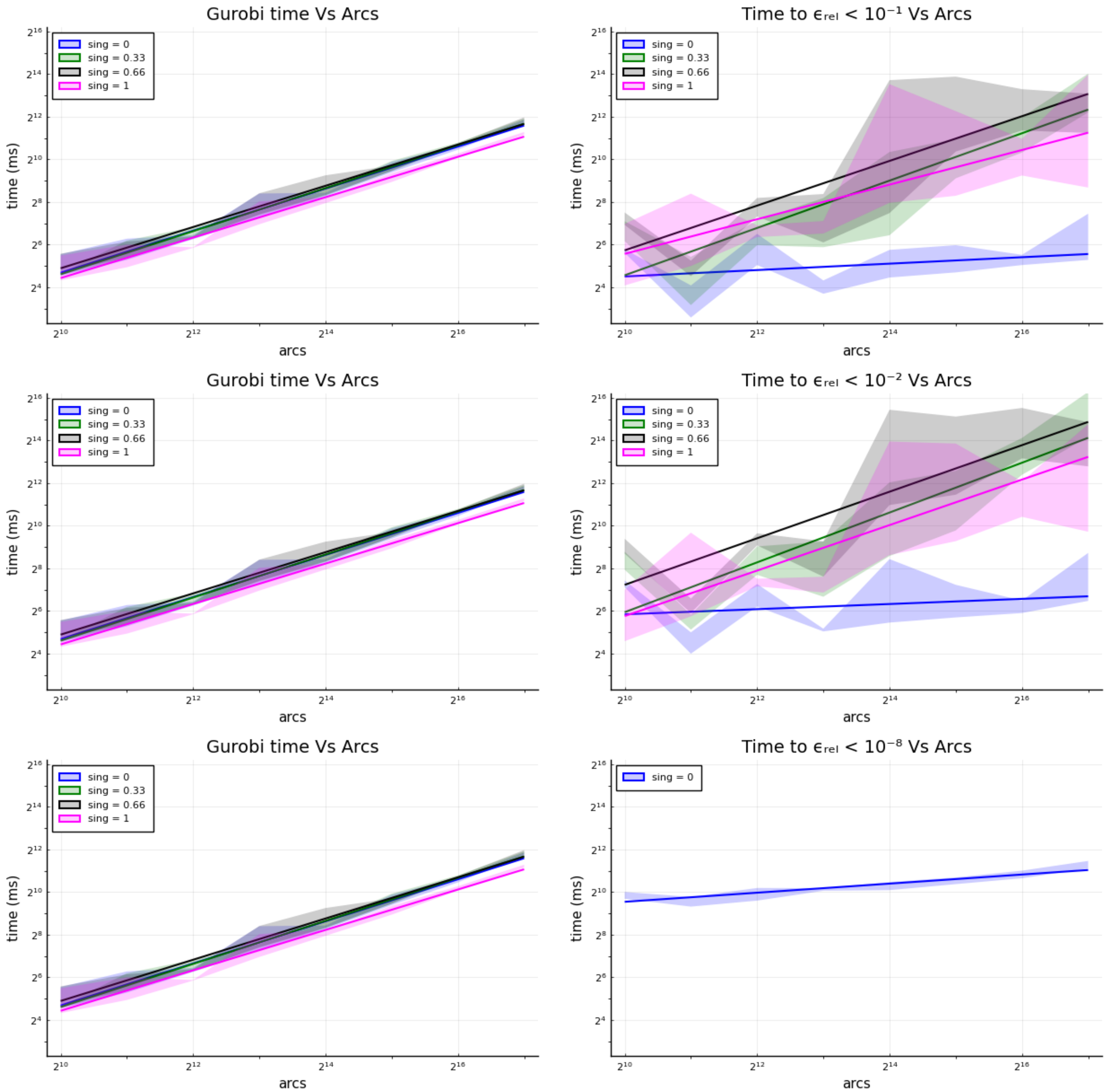
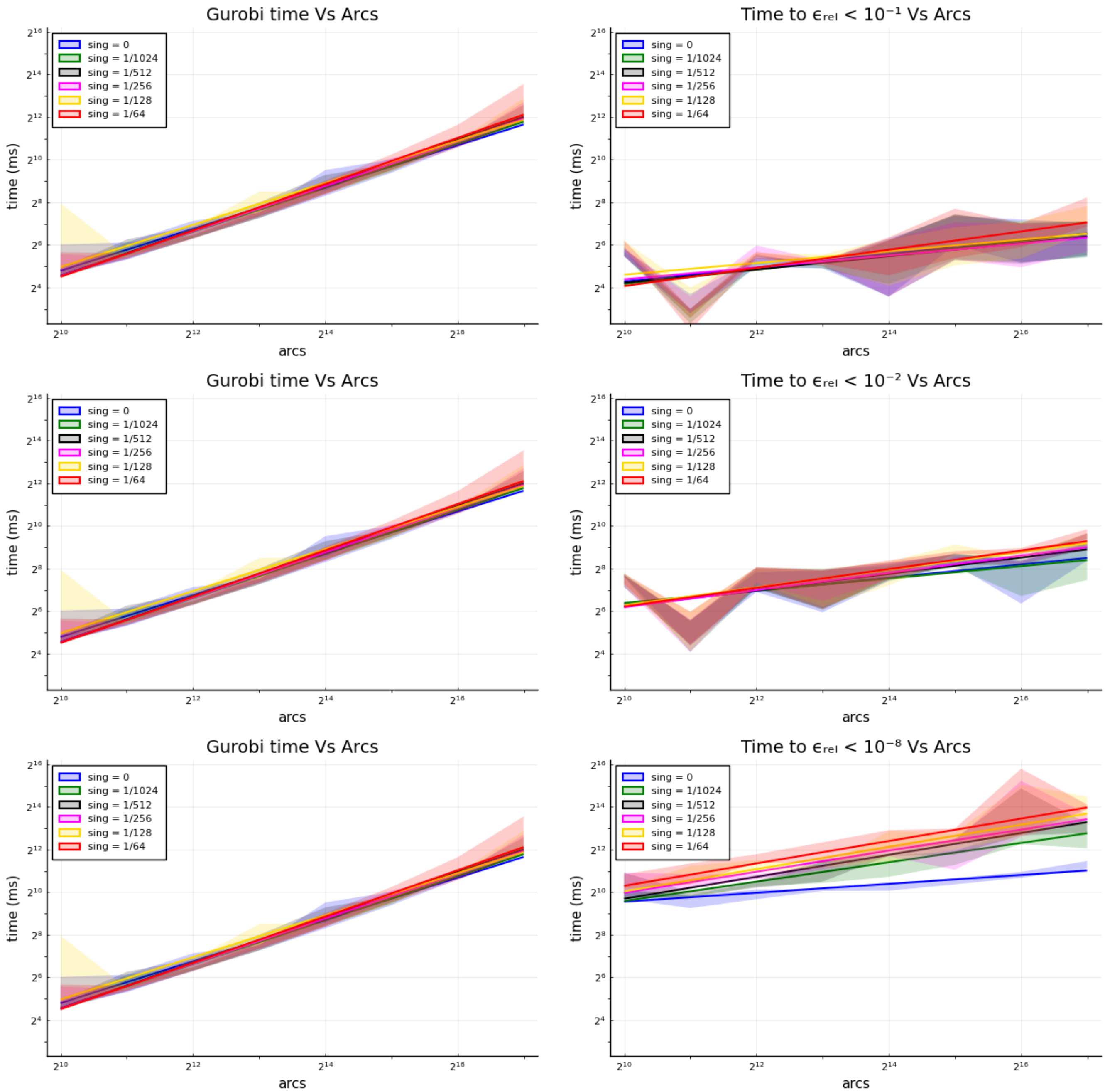


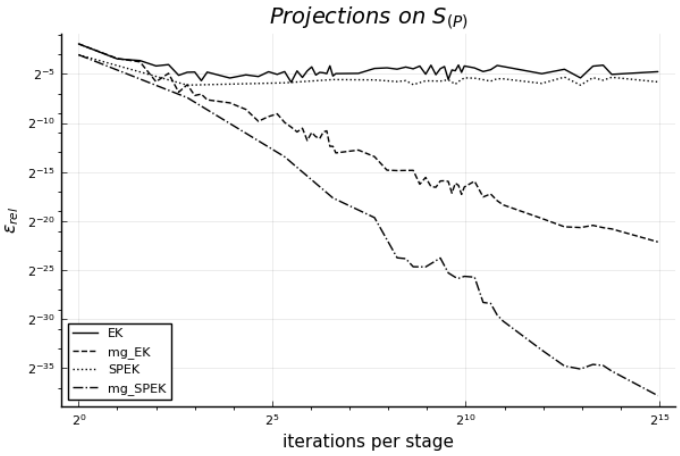
FIGURE 5. ► **Scaling of RNM vs Gurobi for "mildly singular" instances.** The more the required precision the worse is the scaling of RNM; however for mildly singular instances, here up to `sing` = $\frac{1}{64}$, RNM scales better than Gurobi even with 8 digits of precision. A rough analysis suggests that, even for 8 digits of precision, RNM may be considered in place of Gurobi for instances with more than one million of arcs.



Heuristics ▷ The theoretical arguments of §2, about the quality of the many different possible projections on the feasible space $S_{(P)}$, have been fully confirmed experimentally. Figure 6 plots the ϵ_{rel}^{gap} against the iterations per stage (where the number of stages is fixed and sufficient to fully exploit the number of iterations per stage). It is easy to check that:

- min-cost max-flow heuristics produce results of higher quality than max-flow ones
- the nearer the starting point to the optimal set, represented in the plot with a higher number of iterations per stage, the more the pre-projection with min-norm subgradient, as per equation 2, is relevant.

FIGURE 6. **Efficacy of projections on the feasible space** $S_{(P)}$. `EK`: EdmondKarp (max-flow), `SPEK`: ShortestPath EdmondKarp (min-cost max-flow), `mg`: pre-projection with min-norm approximate subgradient. `singular`=0.5, `active`=0.87.



We did not analyse the performance in terms of computational efficacy, since all the algorithms in question are well studied and we chose to implement the simplest algorithms (Edmond-Karp for max-flow, Edmond-Karp with shortest path for the min-cost max-flow), which are not the most performant. It could be interesting to evaluate the computational efficiency of a cost/capacity-scaling min-cost max-flow heuristic also as a stopping criterion, e.g. at each restart of the algorithm.
For a complete experimental evaluation, a performance oriented implementation of the algorithm of choice is necessary. That would allow for a completely fair comparison of the efficiency of the methods against external solvers as Gurobi.

Epilogue

We analysed and specialized dual subgradient methods to convex quadratic separable min-cost flow boxed problems. Because of the geometry of the Lagrangian dual, restarted methods have a *raison d'être* and look appealing, among subgradient methods, since they reliably produce high quality solutions, as confirmed by parameter tuning investigations. Experimentally we attained good performance with the RestartedNesterovMomentum with high momentum, for which a proof of convergence is missing. The overall result of this investigation is that the proposed method is advantageous in terms of implementation and parallelization simplicity, memory usage, but also performance for mildy-singular, big enough, instances.
The existence of a cheap tuner could lead to tremendous performance improvements in the least singular instances.
From each point in the dual space we can calculate an approximate, almost feasible, point in the primal space via the ad-hoc projections on the feasible space $S_{(P)}$; an exact bounding interval for the optimal objective value is finally calculated through the error analysis illustrated in §2.
Of great help to the fast-paced investigation was the flexibility of both the Julia package we developed and the Julia language itself.

Gruppo 34
Miccinesi M., michele.miccinesi@gmail.com
Big thanks to **Lapo Toloni**, coauthor of this project - until his master degree got too near. ☺

Bibliography

[1] Frangioni A., Gendron B., Gorgone E., "On the Computational Efficiency of Subgradient Methods: a Case Study with Lagrangian Bounds", *Mathematical Programming Computation*, **9**:4 (2017), 573–604.

[2] D'Antonio G., Frangioni A., "Convergence Analysis of Deflected Conditional Approximate Subgradient Methods", *SIAM Journal of Optimization*, **20**:1 (2009), 357-386.

[3] Anstreicher K. M., Wolsey L. A., "Two "well-known" properties of subgradient optimization", *Mathematical Programming*, **120** (2009), 213–220.

[4] Nesterov Y., "Lectures on convex optimization", 2018.

[5] Gustavsson E., Patriksson M., Strömberg A., "Primal convergence from dual subgradient methods for convex optimization", *Mathematical Programming*, **150** (2014).

[6] Önnheim M., Gustavsson E., Strömberg A. B., Patriksson M., Larsson T., "Ergodic, primal convergence in dual subgradient schemes for convex programming, II: the case of inconsistent primal problems", *Mathematical Programming*, **163** (2017), 57-–84.

[7] Boyd S., "Subgradient Methods", *Notes for EE364b, Stanford University*, 2014.

[8] Bazaraa M. S., Sherali H. D., Shetty C. M., "Nonlinear programming theory and applications", 1993.

[9] Boyd S., Vandenberghe L., "Convex Optimization", 2004.

[10] Nocendal J., Wright S., "Numerical Optimization", 2006.

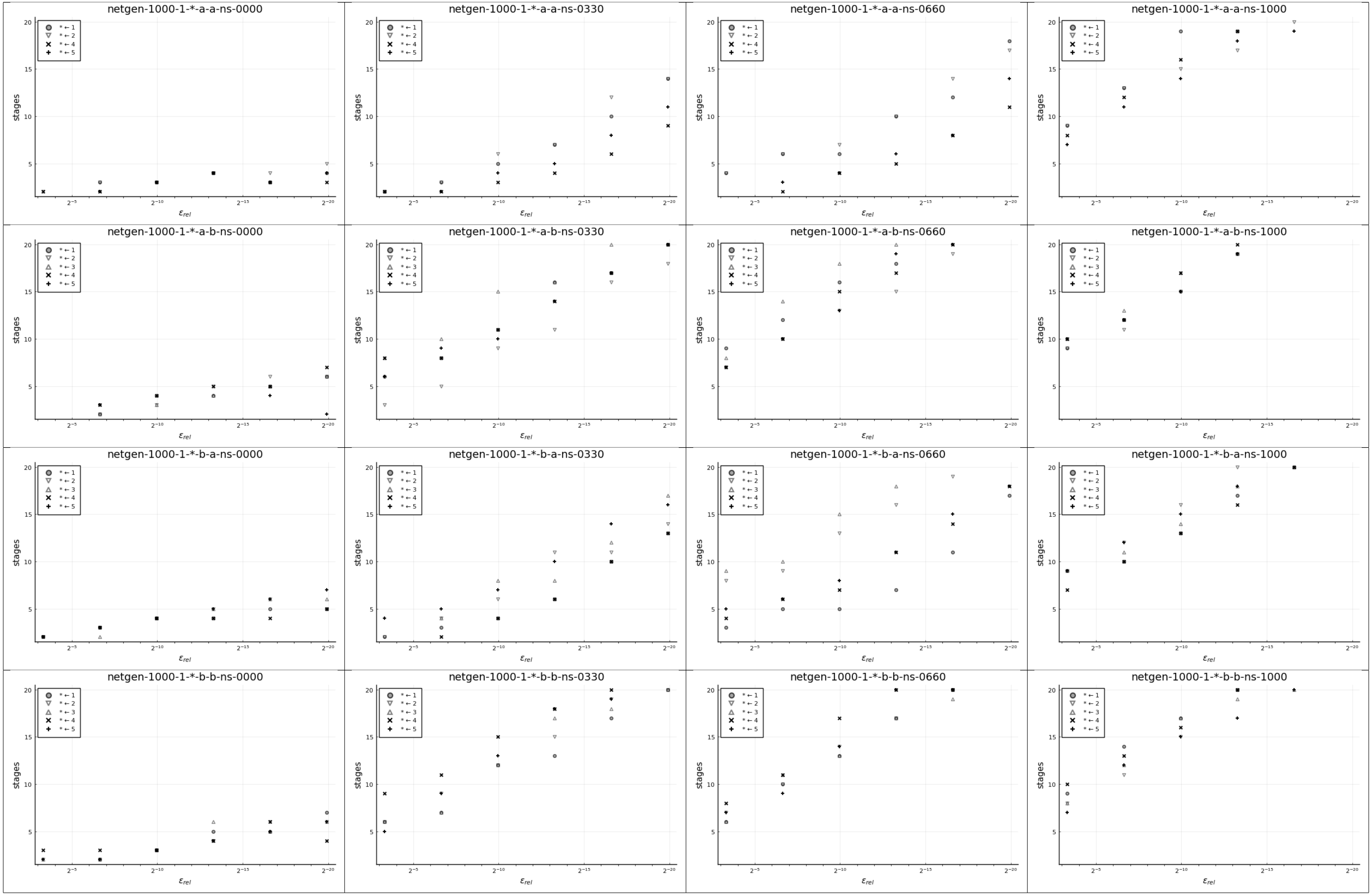
[11] Olsson C., "Lagrangian Heuristics for Strictly Convex Quadratic Minimum Cost Network Flow Problems", 2005.

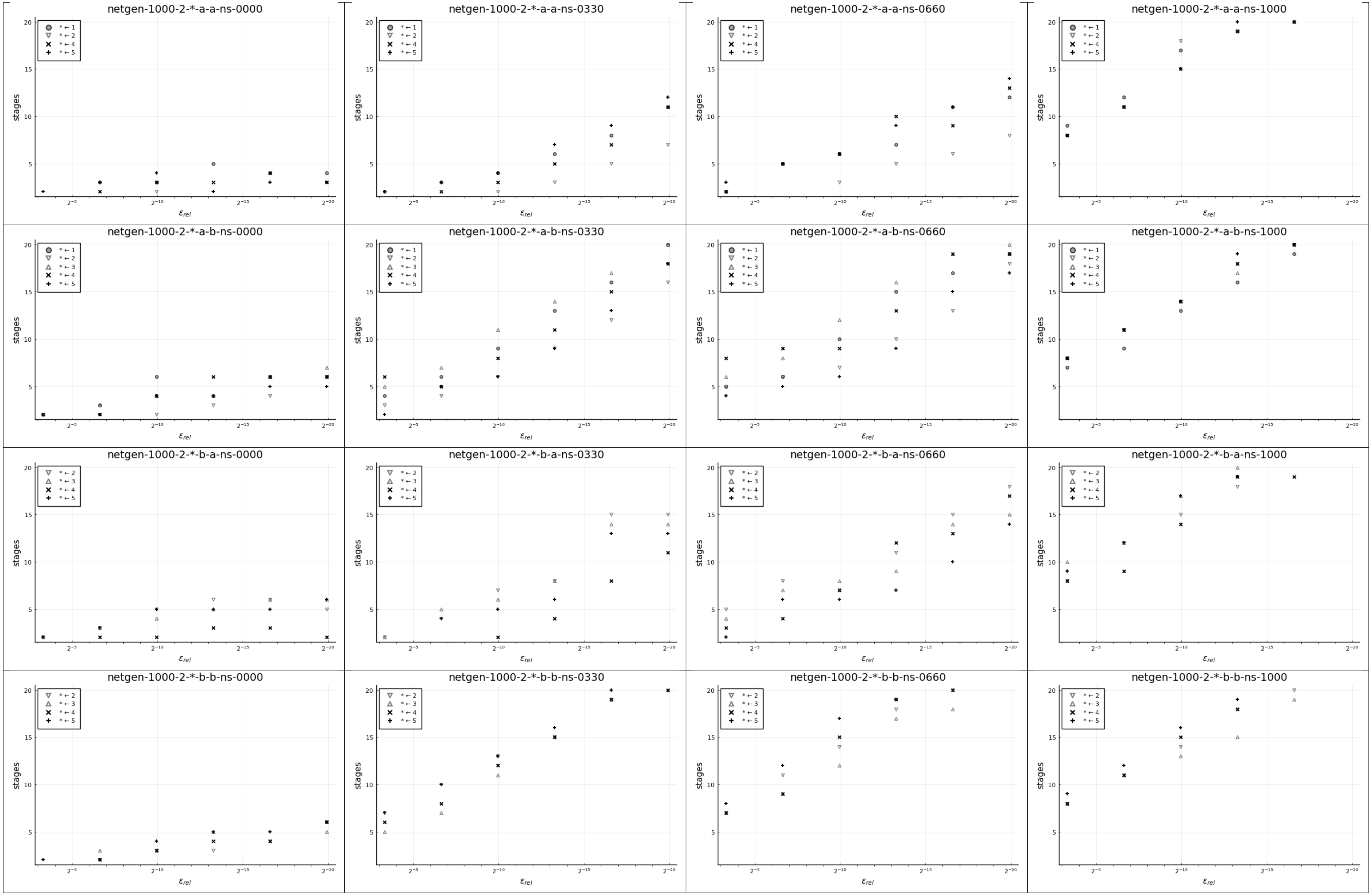
[12] Yang T., Lin Q., "RSG: Beating Subgradient Method without Smoothness and Strong Convexity", 2016.

[13] Duchi J., Hazan E., Singer Y., "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", 2011.

[14] Diakonikolas J., Jordan M. I., "Generalized Momentum-Based Methods: A Hamiltonian Perspective", 2019.

Table 5. Restarted Nesterov Momentum : number of stages against ϵ_{rel}




netgen-1000-2-*-a-a-ns-0000
netgen-1000-2-*-a-a-ns-0330
netgen-1000-2-*-a-a-ns-0660
netgen-1000-2-*-a-a-ns-1000
netgen-1000-2-*-a-b-ns-0000
netgen-1000-2-*-a-b-ns-0330
netgen-1000-2-*-a-b-ns-0660
netgen-1000-2-*-a-b-ns-1000
netgen-1000-2-*-b-a-ns-0000
netgen-1000-2-*-b-a-ns-0330
netgen-1000-2-*-b-a-ns-0660
netgen-1000-2-*-b-a-ns-1000
netgen-1000-2-*-b-b-ns-0000
netgen-1000-2-*-b-b-ns-0330
netgen-1000-2-*-b-b-ns-0660
netgen-1000-2-*-b-b-ns-1000
stages
ε_{rel}
* ← 1
* ← 2
* ← 3
* ← 4
* ← 5

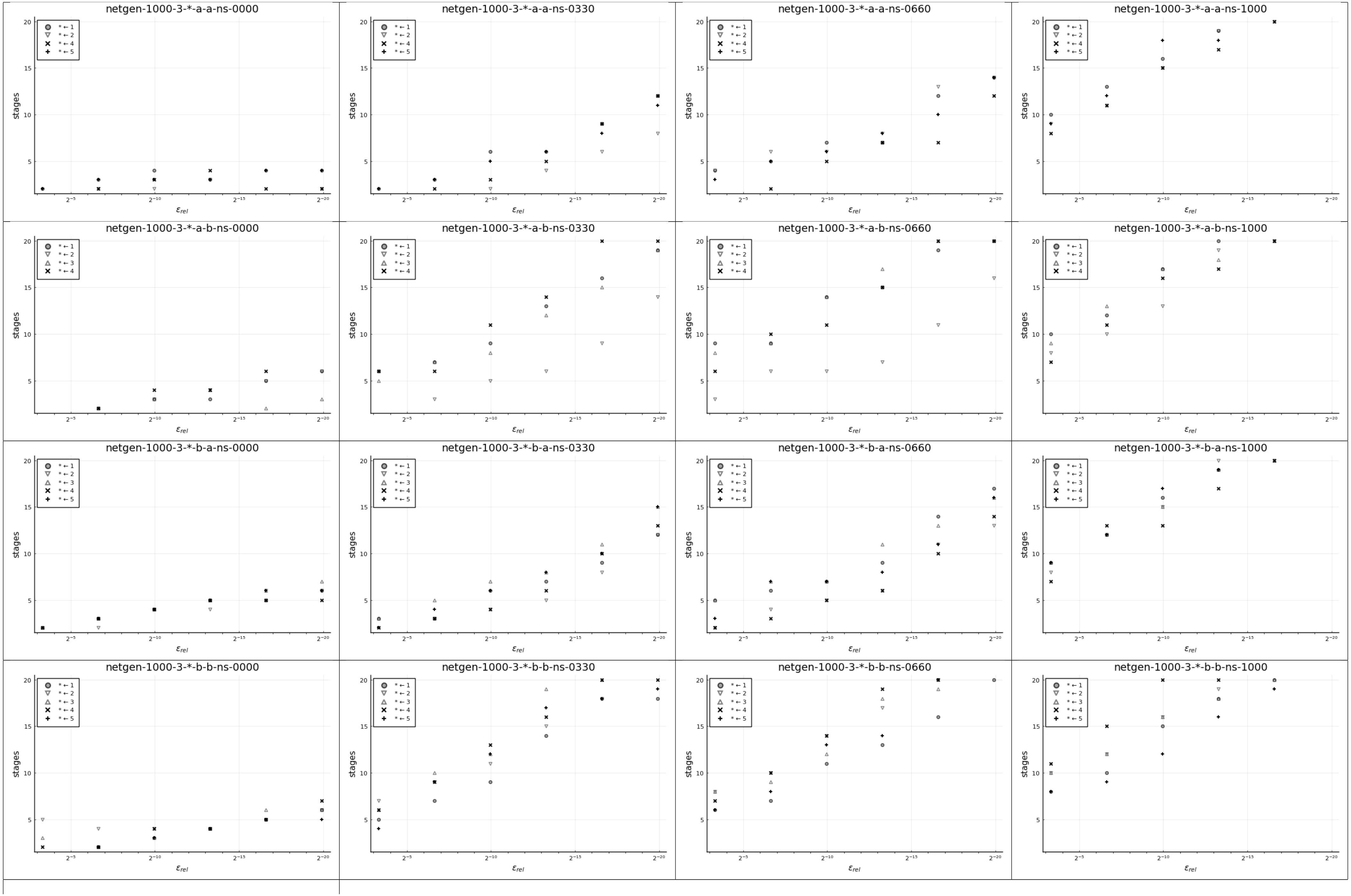
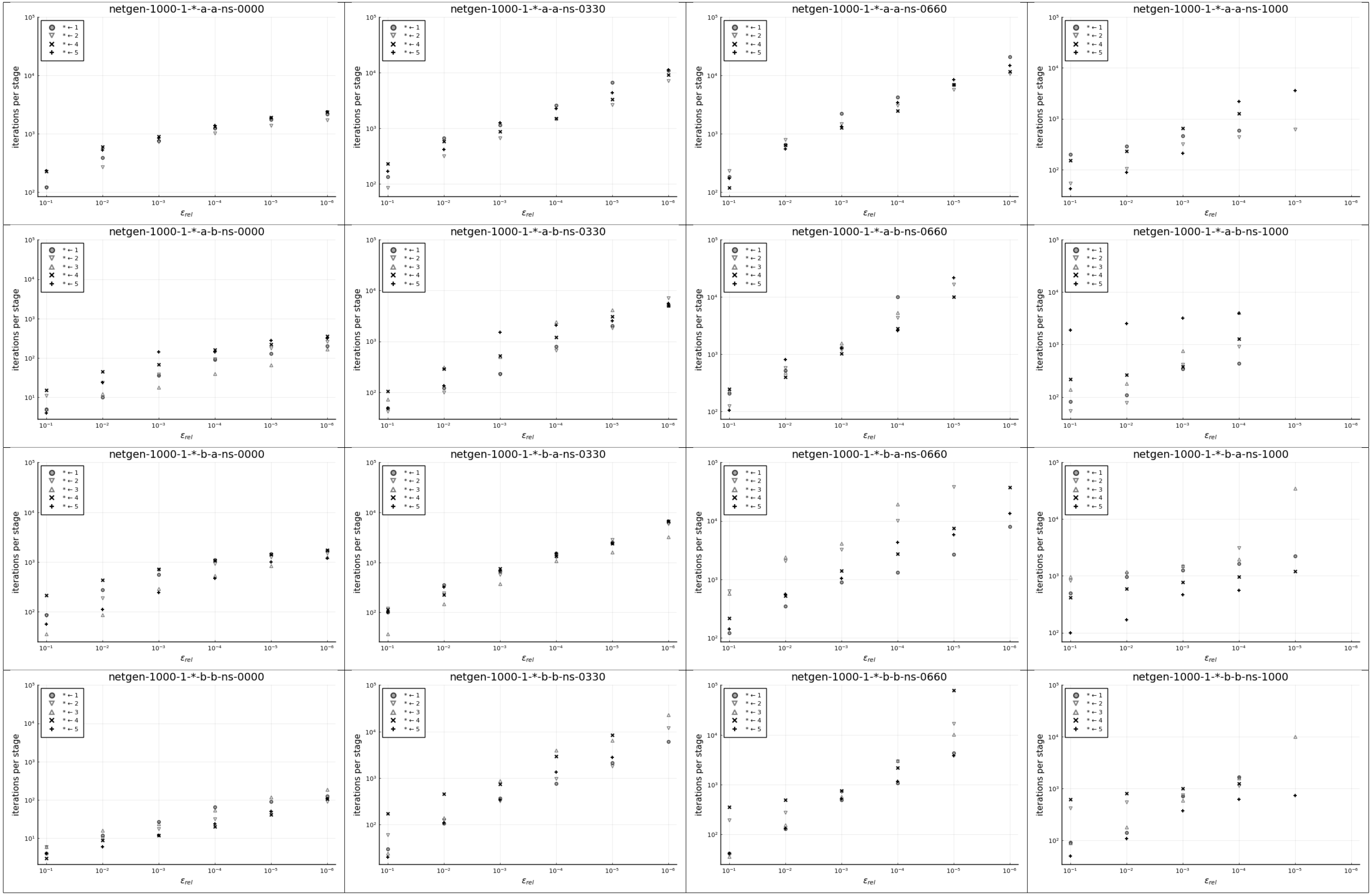
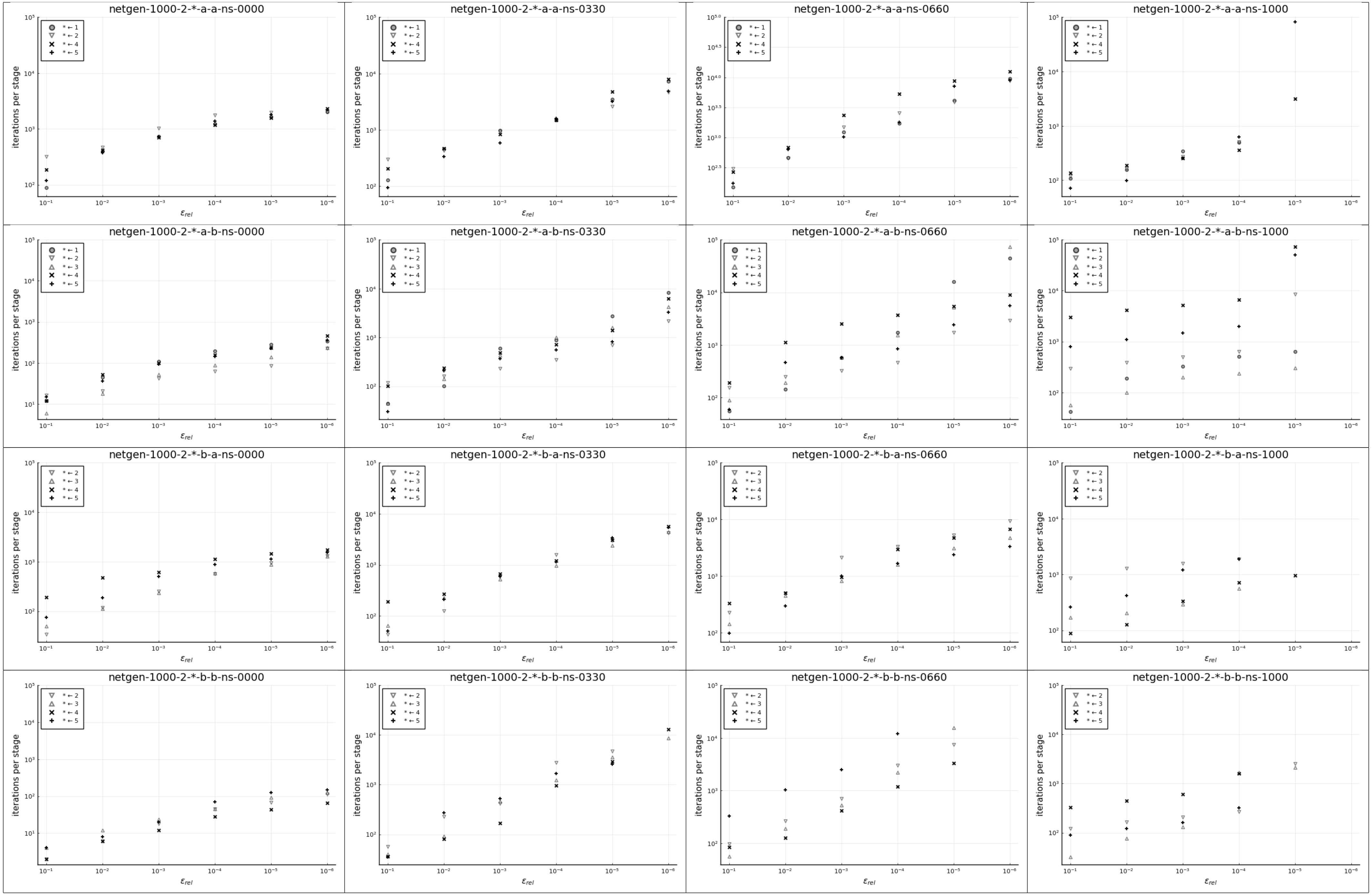
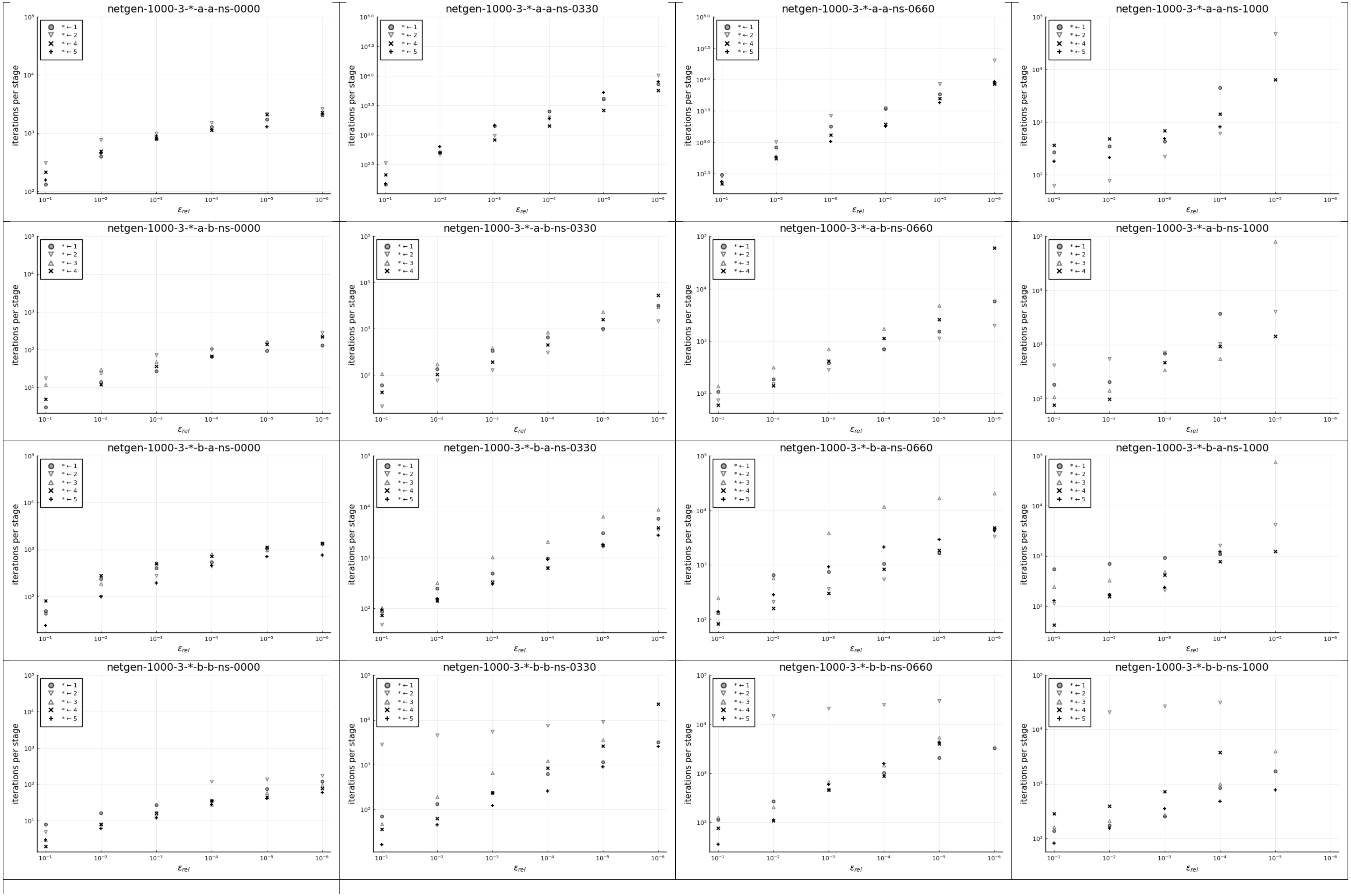

netgen-1000-3-*-a-a-ns-0000
netgen-1000-3-*-a-a-ns-0330
netgen-1000-3-*-a-a-ns-0660
netgen-1000-3-*-a-a-ns-1000
netgen-1000-3-*-a-b-ns-0000
netgen-1000-3-*-a-b-ns-0330
netgen-1000-3-*-a-b-ns-0660
netgen-1000-3-*-a-b-ns-1000
netgen-1000-3-*-b-a-ns-0000
netgen-1000-3-*-b-a-ns-0330
netgen-1000-3-*-b-a-ns-0660
netgen-1000-3-*-b-a-ns-1000
netgen-1000-3-*-b-b-ns-0000
netgen-1000-3-*-b-b-ns-0330
netgen-1000-3-*-b-b-ns-0660
netgen-1000-3-*-b-b-ns-1000
stages
ε_{rel}

Table 6. Restarted Nesterov Momentum : number of iterations per stage against ϵ_{rel}




netgen-1000-2-*-a-a-ns-0000
netgen-1000-2-*-a-a-ns-0330
netgen-1000-2-*-a-a-ns-0660
netgen-1000-2-*-a-a-ns-1000
netgen-1000-2-*-a-b-ns-0000
netgen-1000-2-*-a-b-ns-0330
netgen-1000-2-*-a-b-ns-0660
netgen-1000-2-*-a-b-ns-1000
netgen-1000-2-*-b-a-ns-0000
netgen-1000-2-*-b-a-ns-0330
netgen-1000-2-*-b-a-ns-0660
netgen-1000-2-*-b-a-ns-1000
netgen-1000-2-*-b-b-ns-0000
netgen-1000-2-*-b-b-ns-0330
netgen-1000-2-*-b-b-ns-0660
netgen-1000-2-*-b-b-ns-1000
iterations per stage
ε_{rel}


netgen-1000-3-*-a-a-ns-0000
netgen-1000-3-*-a-a-ns-0330
netgen-1000-3-*-a-a-ns-0660
netgen-1000-3-*-a-a-ns-1000
netgen-1000-3-*-a-b-ns-0000
netgen-1000-3-*-a-b-ns-0330
netgen-1000-3-*-a-b-ns-0660
netgen-1000-3-*-a-b-ns-1000
netgen-1000-3-*-b-a-ns-0000
netgen-1000-3-*-b-a-ns-0330
netgen-1000-3-*-b-a-ns-0660
netgen-1000-3-*-b-a-ns-1000
netgen-1000-3-*-b-b-ns-0000
netgen-1000-3-*-b-b-ns-0330
netgen-1000-3-*-b-b-ns-0660
netgen-1000-3-*-b-b-ns-1000
iterations per stage
ε_{rel}

Instance	RSG						RNM					
	mg EK		mg SPEK		Max iter	Iter	mg EK		mg SPEK		Max iter	Iter
	ϵ_{rel}	$\Delta\epsilon_{rel}$	ϵ_{rel}	$\Delta\epsilon_{rel}$			ϵ_{rel}	$\Delta\epsilon_{rel}$	ϵ_{rel}	$\Delta\epsilon_{rel}$		
1000-1-1-a-a-ns-0000	4e-06	2e-12	6e-07	2e-12	7e+04	1e+05	3e-07	4e-12	3e-10	5e-12	1e+03	2e+03
1000-1-1-a-a-ns-0330							2e-05	4e-12	4e-07	6e-12	1e+03	1e+04
1000-1-1-a-a-ns-0660												
1000-1-1-a-a-ns-1000	2e-11	6e-12	2e-11	1e-11	5e+02	1e+04	3e-11	6e-12	3e-11	1e-11	6e+01	2e+03
1000-1-1-a-b-ns-0000	7e-07	2e-12	7e-09	2e-12	1e+02	1e+02	5e-07	1e-12	1e-09	2e-12	6e+01	2e+02
1000-1-1-a-b-ns-0330	1e-05	2e-12	6e-07	3e-12	2e+04	3e+05	8e-06	1e-12	3e-07	3e-12	1e+03	2e+04
1000-1-1-a-b-ns-0660												
1000-1-1-a-b-ns-1000	7e-12	2e-12	7e-12	2e-12	1e+02	3e+03	1e-11	2e-12	1e-11	2e-12	6e+01	2e+03
1000-1-1-b-a-ns-0000	2e-06	2e-12	7e-07	2e-12	2e+04	4e+04	3e-06	3e-12	1e-09	2e-12	5e+02	2e+03
1000-1-1-b-a-ns-0330	3e-06	5e-12	2e-08	5e-12	3e+04	3e+05	4e-05	5e-12	2e-07	5e-12	5e+02	6e+03
1000-1-1-b-a-ns-0660	3e-06	2e-12	6e-08	4e-12	3e+04	4e+05	3e-06	6e-12	2e-08	6e-12	4e+03	5e+04
1000-1-1-b-a-ns-1000	8e-12	6e-12	8e-12	4e-12	1e+03	2e+04	7e-12	6e-12	7e-12	4e-12	3e+02	6e+03
1000-1-1-b-b-ns-0000	2e-06	4e-12	5e-09	6e-12	1e+02	1e+02	9e-08	2e-12	1e-11	2e-12	6e+01	1e+02
1000-1-1-b-b-ns-0330	4e-05	3e-12	7e-07	6e-12	4e+03	7e+04	5e-06	2e-12	7e-09	4e-12	2e+03	4e+04
1000-1-1-b-b-ns-0660	5e-05	1e-12	7e-07	2e-12	2e+04	3e+05	2e-05	1e-12	4e-07	2e-12	2e+03	4e+04
1000-1-1-b-b-ns-1000	3e-11	6e-12	3e-11	3e-12	2e+03	5e+04	2e-11	6e-12	2e-11	3e-12	5e+02	1e+04
1000-1-2-a-a-ns-0000	8e-08	2e-12	7e-10	9e-13	3e+04	4e+04	5e-06	2e-12	7e-08	2e-12	5e+02	2e+03
1000-1-2-a-a-ns-0330	4e-06	3e-12	4e-07	3e-12	7e+04	7e+05	7e-06	2e-12	7e-08	3e-12	1e+03	1e+04
1000-1-2-a-a-ns-0660	7e-06	5e-12	1e-07	6e-12	7e+04	9e+05	3e-06	5e-12	1e-08	5e-12	2e+04	2e+05
1000-1-2-a-a-ns-1000	5e-12	1e-11	4e-12	2e-12	3e+02	6e+03	1e-11	1e-11	1e-11	2e-12	6e+01	1e+03
1000-1-2-a-b-ns-0000	3e-06	8e-13	1e-06	1e-12	3e+02	2e+03	3e-05	2e-12	6e-07	1e-12	6e+01	4e+02
1000-1-2-a-b-ns-0330	3e-06	1e-12	9e-07	2e-12	8e+03	1e+05	3e-06	2e-12	5e-07	2e-12	1e+03	2e+04
1000-1-2-a-b-ns-0660												
1000-1-2-a-b-ns-1000	7e-12	8e-12	7e-12	3e-12	3e+02	6e+03	1e-11	8e-12	1e-11	3e-12	1e+02	3e+03
1000-1-2-b-a-ns-0000	6e-06	3e-12	5e-07	2e-12	2e+04	4e+04	3e-07	4e-12	4e-10	5e-12	5e+02	2e+03
1000-1-2-b-a-ns-0330	1e-05	3e-12	9e-08	3e-12	3e+04	4e+05	9e-06	3e-12	1e-07	3e-12	2e+03	3e+04
1000-1-2-b-a-ns-0660												
1000-1-2-b-a-ns-1000	6e-03	3e-12	2e-07	6e-12	4e+03	9e+04	2e-11	4e-12	2e-11	3e-12	3e+02	6e+03
1000-1-2-b-b-ns-0000	2e-06	1e-12	2e-08	9e-13	6e+01	1e+02	4e-09	2e-12	1e-14	2e-12	6e+01	1e+02
1000-1-2-b-b-ns-0330	5e-05	2e-12	6e-07	1e-12	8e+03	2e+05	6e-05	2e-12	1e-06	2e-12	3e+02	7e+03
1000-1-2-b-b-ns-0660	2e-05	4e-12	1e-07	2e-12	3e+04	7e+05	4e-05	2e-12	4e-07	2e-12	4e+03	9e+04
1000-1-2-b-b-ns-1000	2e-07	7e-13	1e-08	2e-12	5e+02	1e+04	1e-11	2e-12	1e-11	3e-12	1e+02	3e+03
1000-1-3-a-a-ns-0000												
1000-1-3-a-a-ns-0330												
1000-1-3-a-a-ns-0660												
1000-1-3-a-a-ns-1000												
1000-1-3-a-b-ns-0000	4e-08	8e-13	8e-11	1e-12	1e+02	1e+02	2e-08	2e-12	1e-11	2e-12	6e+01	1e+02

1000-1-3-a-b-ns-0330	4e-06	3e-12	4e-08	4e-12	7e+04	1e+06	1e-05	1e-12	2e-07	4e-12	2e+03	5e+04
1000-1-3-a-b-ns-0660	7e-06	5e-12	1e-07	5e-12	7e+04	1e+06	1e-05	3e-12	7e-07	3e-12	2e+04	4e+05
1000-1-3-a-b-ns-1000	2e-11	3e-12	2e-11	4e-12	4e+03	1e+05	3e-11	3e-12	3e-11	4e-12	3e+02	6e+03
1000-1-3-b-a-ns-0000	1e-06	2e-12	2e-08	3e-12	8e+03	9e+03	3e-05	2e-12	3e-07	2e-12	3e+02	1e+03
1000-1-3-b-a-ns-0330	2e-06	2e-12	5e-07	2e-12	8e+03	1e+05	1e-06	1e-12	1e-07	2e-12	2e+03	3e+04
1000-1-3-b-a-ns-0660												
1000-1-3-b-a-ns-1000	4e-12	2e-12	4e-12	2e-12	1e+03	2e+04	1e-11	2e-12	1e-11	2e-12	1e+02	3e+03
1000-1-3-b-b-ns-0000	3e-05	2e-12	6e-07	4e-12	1e+02	4e+02	2e-06	3e-12	7e-10	5e-12	6e+01	2e+02
1000-1-3-b-b-ns-0330	5e-05	4e-12	7e-07	3e-12	8e+03	1e+05	8e-05	1e-12	5e-07	3e-12	4e+03	8e+04
1000-1-3-b-b-ns-0660	1e-05	1e-12	2e-07	3e-12	7e+04	1e+06	4e-05	1e-12	3e-07	3e-12	4e+03	9e+04
1000-1-3-b-b-ns-1000	4e-12	6e-12	4e-12	5e-12	2e+03	4e+04	1e-11	6e-12	1e-11	5e-12	3e+02	6e+03
1000-1-4-a-a-ns-0000							3e-06	1e-12	1e-08	2e-12	1e+03	3e+03
1000-1-4-a-a-ns-0330							5e-07	3e-12	5e-09	3e-12	2e+03	2e+04
1000-1-4-a-a-ns-0660							1e-06	8e-12	4e-09	8e-12	2e+03	2e+04
1000-1-4-a-a-ns-1000	9e-11	8e-12	9e-11	8e-12	4e+03	9e+04	1e-11	8e-12	1e-11	8e-12	1e+03	2e+04
1000-1-4-a-b-ns-0000	9e-07	2e-12	4e-08	2e-12	5e+02	1e+03	5e-05	2e-12	1e-06	3e-12	6e+01	7e+02
1000-1-4-a-b-ns-0330	2e-05	3e-12	2e-07	3e-12	3e+04	6e+05	1e-05	1e-12	2e-07	2e-12	1e+03	2e+04
1000-1-4-a-b-ns-0660	3e-05	4e-12	8e-07	6e-12	3e+04	7e+05	2e-05	3e-12	1e-07	3e-12	8e+03	2e+05
1000-1-4-a-b-ns-1000	1e-03	1e-12	1e-06	3e-13	5e+02	1e+04	1e-11	2e-12	1e-11	5e-15	5e+02	1e+04
1000-1-4-b-a-ns-0000	4e-08	3e-12	3e-11	2e-12	7e+04	7e+04	2e-07	4e-12	1e-08	5e-12	5e+02	2e+03
1000-1-4-b-a-ns-0330	2e-06	2e-12	1e-08	2e-12	7e+04	7e+05	5e-06	2e-12	5e-08	3e-12	1e+03	1e+04
1000-1-4-b-a-ns-0660	7e-07	3e-12	1e-07	4e-12	7e+04	9e+05	9e-06	6e-12	5e-08	4e-12	2e+04	3e+05
1000-1-4-b-a-ns-1000	4e-12	1e-12	4e-12	8e-13	3e+02	5e+03	6e-12	1e-12	5e-12	8e-13	1e+02	3e+03
1000-1-4-b-b-ns-0000	9e-07	1e-12	3e-09	1e-12	6e+01	1e+02	2e-09	2e-12	3e-15	2e-12	6e+01	2e+02
1000-1-4-b-b-ns-0330	4e-05	1e-12	3e-07	2e-12	8e+03	2e+05	4e-05	2e-12	6e-07	4e-12	1e+03	2e+04
1000-1-4-b-b-ns-0660	3e-05	2e-12	4e-07	2e-12	8e+03	2e+05	4e-05	4e-12	7e-07	3e-12	1e+03	3e+04
1000-1-4-b-b-ns-1000	3e-11	3e-12	3e-11	1e-12	4e+03	1e+05	3e-03	6e-12	6e-07	5e-12	1e+02	3e+03
1000-1-5-a-a-ns-0000							3e-05	3e-12	4e-09	3e-12	1e+03	3e+03
1000-1-5-a-a-ns-0330							6e-06	4e-12	2e-08	4e-12	2e+03	2e+04
1000-1-5-a-a-ns-0660							9e-07	1e-12	5e-08	3e-12	2e+04	2e+05
1000-1-5-a-a-ns-1000	5e-10	5e-12	5e-10	3e-12	4e+03	8e+04	3e-12	5e-12	3e-12	3e-12	5e+02	1e+04
1000-1-5-a-b-ns-0000	8e-06	3e-12	5e-07	5e-12	1e+02	4e+02	1e-05	1e-12	1e-07	2e-12	6e+01	3e+02
1000-1-5-a-b-ns-0330	3e-05	5e-12	3e-07	5e-12	2e+04	3e+05	2e-05	2e-12	2e-07	4e-12	5e+02	1e+04
1000-1-5-a-b-ns-0660	4e-06	4e-12	3e-08	4e-12	7e+04	1e+06						
1000-1-5-a-b-ns-1000	1e-11	3e-12	1e-11	5e-12	8e+03	2e+05	3e-11	3e-12	3e-11	2e-12	3e+02	6e+03
1000-1-5-b-a-ns-0000	4e-06	3e-12	4e-07	3e-12	4e+03	9e+03	3e-06	1e-12	1e-07	2e-12	3e+02	1e+03
1000-1-5-b-a-ns-0330	4e-05	4e-12	2e-07	4e-12	2e+04	2e+05	2e-05	3e-12	4e-07	5e-12	5e+02	8e+03
1000-1-5-b-a-ns-0660	3e-06	4e-12	8e-07	6e-12	3e+04	6e+05	3e-06	4e-12	2e-07	5e-12	2e+03	3e+04
1000-1-5-b-a-ns-1000	9e-12	7e-12	9e-12	6e-12	3e+02	6e+03	2e-11	7e-12	2e-11	6e-12	6e+01	1e+03
1000-1-5-b-b-ns-0000	3e-05	2e-12	1e-07	3e-12	6e+01	2e+02	3e-08	3e-12	5e-14	4e-12	6e+01	1e+02
1000-1-5-b-b-ns-0330	1e-04	1e-12	8e-07	4e-12	8e+03	2e+05	7e-05	2e-12	3e-07	3e-12	1e+03	2e+04

1000-1-5-b-b-ns-0660	4e-05	1e-12	8e-08	5e-12	2e+04	3e+05	4e-06	3e-12	3e-09	5e-12	8e+03	2e+05
1000-1-5-b-b-ns-1000	2e-06	8e-13	3e-12	1e-13	3e+02	5e+03	9e-12	2e-12	9e-12	4e-12	6e+01	1e+03
1000-2-1-a-a-ns-0000	5e-06	4e-12	3e-07	5e-12	7e+04	1e+05	2e-06	5e-12	1e-07	4e-12	5e+02	2e+03
1000-2-1-a-a-ns-0330	3e-07	5e-12	9e-07	4e-12	7e+04	7e+05	7e-07	4e-12	2e-09	4e-12	1e+03	1e+04
1000-2-1-a-a-ns-0660	2e-06	4e-12	2e-07	5e-12	7e+04	7e+05	4e-07	6e-12	8e-09	7e-12	2e+03	2e+04
1000-2-1-a-a-ns-1000	1e-11	5e-12	1e-11	4e-12	5e+02	1e+04	3e-11	5e-12	3e-11	4e-12	6e+01	2e+03
1000-2-1-a-b-ns-0000	2e-07	2e-12	7e-10	2e-12	5e+02	5e+02	1e-06	3e-12	3e-10	4e-12	1e+02	4e+02
1000-2-1-a-b-ns-0330	2e-05	3e-12	6e-07	4e-12	2e+04	3e+05	1e-05	4e-12	4e-07	2e-12	1e+03	2e+04
1000-2-1-a-b-ns-0660	2e-05	3e-12	2e-07	4e-12	2e+04	3e+05	4e-06	4e-13	7e-09	3e-12	4e+03	7e+04
1000-2-1-a-b-ns-1000	2e-12	3e-12	2e-12	3e-12	3e+02	5e+03	4e-12	3e-12	3e-12	3e-12	6e+01	1e+03
1000-2-1-b-a-ns-0000												
1000-2-1-b-a-ns-0330												
1000-2-1-b-a-ns-0660												
1000-2-1-b-a-ns-1000												
1000-2-1-b-b-ns-0000												
1000-2-1-b-b-ns-0330												
1000-2-1-b-b-ns-0660												
1000-2-1-b-b-ns-1000												
1000-2-2-a-a-ns-0000							1e-09	1e-12	3e-12	1e-12	2e+03	2e+03
1000-2-2-a-a-ns-0330							3e-08	2e-12	1e-08	2e-12	2e+03	1e+04
1000-2-2-a-a-ns-0660							7e-07	2e-12	8e-07	1e-12	2e+03	2e+04
1000-2-2-a-a-ns-1000	7e-12	9e-13	7e-12	6e-13	3e+02	6e+03	2e-11	9e-13	2e-11	6e-13	6e+01	1e+03
1000-2-2-a-b-ns-0000	6e-07	9e-13	9e-08	1e-12	1e+03	3e+03	2e-06	6e-13	2e-09	7e-13	6e+01	3e+02
1000-2-2-a-b-ns-0330	1e-06	1e-12	8e-08	1e-12	4e+03	5e+04	5e-06	1e-12	6e-07	1e-12	3e+02	4e+03
1000-2-2-a-b-ns-0660	4e-06	1e-12	3e-07	2e-12	8e+03	1e+05	2e-05	4e-13	2e-07	9e-13	1e+03	2e+04
1000-2-2-a-b-ns-1000	6e-12	8e-12	6e-12	5e-12	5e+02	1e+04	1e-11	8e-12	1e-11	5e-12	6e+01	1e+03
1000-2-2-b-a-ns-0000	5e-08	2e-12	2e-10	2e-12	8e+03	7e+03	1e-05	1e-12	4e-08	1e-12	3e+02	1e+03
1000-2-2-b-a-ns-0330	1e-05	2e-12	5e-07	4e-12	2e+04	2e+05	1e-05	3e-12	2e-07	4e-12	5e+02	7e+03
1000-2-2-b-a-ns-0660	4e-05	5e-12	1e-06	6e-12	2e+04	4e+05	5e-06	9e-13	4e-07	2e-12	5e+02	9e+03
1000-2-2-b-a-ns-1000	9e-12	6e-12	9e-12	4e-12	5e+02	1e+04	2e-11	2e-12	2e-11	6e-12	1e+02	3e+03
1000-2-2-b-b-ns-0000	4e-05	1e-12	8e-07	4e-12	6e+01	2e+02	4e-07	3e-12	2e-10	2e-12	6e+01	1e+02
1000-2-2-b-b-ns-0330	4e-05	2e-12	3e-07	5e-12	3e+04	6e+05	6e-05	2e-12	6e-07	5e-12	1e+03	2e+04
1000-2-2-b-b-ns-0660	1e-05	6e-12	2e-08	4e-12	3e+04	7e+05	5e-05	2e-12	9e-07	4e-12	2e+03	5e+04
1000-2-2-b-b-ns-1000	1e-10	8e-12	1e-10	7e-12	2e+03	5e+04	5e-03	8e-13	4e-07	0e+00	1e+02	3e+03
1000-2-3-a-a-ns-0000												
1000-2-3-a-a-ns-0330												
1000-2-3-a-a-ns-0660												
1000-2-3-a-a-ns-1000												
1000-2-3-a-b-ns-0000	1e-06	3e-12	5e-09	2e-12	3e+02	2e+02	7e-06	2e-12	1e-07	1e-12	6e+01	3e+02
1000-2-3-a-b-ns-0330	1e-05	5e-12	2e-07	5e-12	2e+04	3e+05	5e-06	6e-12	4e-08	8e-12	1e+03	2e+04
1000-2-3-a-b-ns-0660	2e-05	4e-12	6e-07	5e-12	2e+04	3e+05	9e-06	6e-12	5e-08	5e-12	1e+03	2e+04

1000-2-3-a-b-ns-1000	4e-12	4e-12	3e-12	1e-11	6e+01	1e+03	5e-12	4e-12	5e-12	1e-11	6e+01	1e+03
1000-2-3-b-a-ns-0000	4e-06	1e-12	1e-07	1e-12	4e+03	7e+03	1e-05	3e-12	5e-07	4e-12	3e+02	2e+03
1000-2-3-b-a-ns-0330	6e-05	2e-12	3e-07	3e-12	8e+03	1e+05	1e-06	4e-12	6e-09	5e-12	5e+02	7e+03
1000-2-3-b-a-ns-0660	9e-06	1e-12	2e-07	3e-12	2e+04	2e+05	5e-06	3e-12	5e-08	3e-12	5e+02	7e+03
1000-2-3-b-a-ns-1000	1e-10	8e-12	1e-10	0e+00	3e+02	6e+03	3e-11	8e-12	3e-11	0e+00	6e+01	2e+03
1000-2-3-b-b-ns-0000	2e-06	3e-12	7e-09	3e-12	1e+02	1e+02	8e-08	2e-12	2e-11	3e-12	6e+01	1e+02
1000-2-3-b-b-ns-0330	6e-05	9e-13	5e-07	3e-12	8e+03	1e+05	8e-05	2e-12	9e-07	3e-12	5e+02	1e+04
1000-2-3-b-b-ns-0660	4e-05	2e-12	7e-07	1e-12	2e+04	3e+05	5e-05	2e-12	7e-07	5e-12	8e+03	2e+05
1000-2-3-b-b-ns-1000	4e-03	5e-12	2e-08	2e-11	2e+03	4e+04	4e-03	5e-12	6e-07	2e-11	1e+02	3e+03
1000-2-4-a-a-ns-0000	3e-05	9e-13	2e-07	2e-12	7e+04	2e+05	9e-07	2e-12	8e-11	2e-12	1e+03	2e+03
1000-2-4-a-a-ns-0330							3e-07	2e-12	5e-10	2e-12	2e+03	2e+04
1000-2-4-a-a-ns-0660							5e-06	2e-12	6e-07	2e-12	2e+04	2e+05
1000-2-4-a-a-ns-1000	5e-12	2e-12	5e-12	3e-12	5e+02	1e+04	9e-12	2e-12	9e-12	3e-12	1e+02	3e+03
1000-2-4-a-b-ns-0000	3e-06	3e-12	2e-07	4e-12	1e+03	2e+03	2e-06	3e-12	2e-08	5e-12	1e+02	5e+02
1000-2-4-a-b-ns-0330	6e-06	1e-12	4e-07	4e-12	4e+03	7e+04	3e-06	7e-12	3e-07	5e-12	5e+02	9e+03
1000-2-4-a-b-ns-0660	8e-07	2e-12	1e-07	2e-12	2e+04	3e+05	4e-06	2e-12	1e-07	3e-12	5e+02	1e+04
1000-2-4-a-b-ns-1000	4e-12	2e-11	4e-12	2e-11	1e+03	2e+04	1e-11	2e-11	1e-11	2e-11	5e+02	1e+04
1000-2-4-b-a-ns-0000							2e-06	2e-12	5e-07	3e-12	5e+02	2e+03
1000-2-4-b-a-ns-0330							7e-05	4e-12	2e-08	5e-12	1e+03	9e+03
1000-2-4-b-a-ns-0660	3e-07	4e-12	2e-07	7e-12	7e+04	9e+05	7e-06	3e-12	6e-07	7e-12	1e+03	2e+04
1000-2-4-b-a-ns-1000	2e-12	3e-12	2e-12	4e-13	5e+02	1e+04	4e-12	2e-12	4e-12	4e-13	6e+01	1e+03
1000-2-4-b-b-ns-0000	1e-07	2e-12	4e-11	2e-12	6e+01	5e+01	2e-09	2e-12	8e-15	1e-12	6e+01	1e+02
1000-2-4-b-b-ns-0330	9e-06	4e-12	3e-07	2e-12	8e+03	2e+05	6e-06	3e-12	9e-08	2e-12	5e+02	1e+04
1000-2-4-b-b-ns-0660	4e-05	1e-12	6e-07	2e-12	8e+03	2e+05	3e-05	4e-12	8e-07	2e-12	5e+02	1e+04
1000-2-4-b-b-ns-1000	9e-03	8e-12	2e-07	5e-12	3e+02	6e+03	1e-11	9e-12	1e-11	6e-12	3e+02	6e+03
1000-2-5-a-a-ns-0000	1e-06	2e-12	1e-07	2e-12	7e+04	1e+05	6e-08	2e-12	5e-12	2e-12	1e+03	2e+03
1000-2-5-a-a-ns-0330	4e-06	3e-12	2e-08	2e-12	7e+04	6e+05	1e-06	5e-12	7e-09	7e-12	1e+03	1e+04
1000-2-5-a-a-ns-0660	2e-06	2e-12	4e-08	2e-12	7e+04	7e+05	3e-05	8e-12	4e-07	9e-12	5e+02	7e+03
1000-2-5-a-a-ns-1000	6e-12	2e-12	6e-12	4e-12	4e+03	9e+04	1e-11	2e-12	1e-11	4e-12	3e+02	6e+03
1000-2-5-a-b-ns-0000	5e-07	4e-13	2e-07	6e-13	1e+03	3e+03	6e-04	9e-13	9e-07	1e-12	6e+01	5e+02
1000-2-5-a-b-ns-0330	1e-06	6e-13	2e-07	8e-13	8e+03	1e+05	7e-07	4e-13	9e-08	6e-13	1e+03	2e+04
1000-2-5-a-b-ns-0660	5e-05	8e-13	1e-07	2e-12	8e+03	1e+05	4e-05	4e-13	7e-07	6e-13	1e+03	2e+04
1000-2-5-a-b-ns-1000	4e-03	4e-12	8e-07	1e-12	2e+03	5e+04	1e-11	1e-12	1e-11	2e-12	5e+02	1e+04
1000-2-5-b-a-ns-0000	1e-06	3e-12	1e-07	3e-12	2e+04	3e+04	7e-07	2e-12	1e-10	3e-12	5e+02	2e+03
1000-2-5-b-a-ns-0330	2e-06	2e-12	3e-07	2e-12	2e+04	2e+05	1e-06	4e-12	5e-09	2e-12	1e+03	1e+04
1000-2-5-b-a-ns-0660	8e-07	8e-13	1e-07	9e-13	3e+04	3e+05	2e-06	4e-12	2e-08	5e-12	1e+03	1e+04
1000-2-5-b-a-ns-1000	4e-12	7e-13	4e-12	2e-12	5e+02	1e+04	2e-11	7e-13	2e-11	2e-12	1e+02	3e+03
1000-2-5-b-b-ns-0000	2e-06	2e-12	1e-08	3e-12	6e+01	2e+02	9e-08	3e-12	3e-11	2e-12	6e+01	2e+02
1000-2-5-b-b-ns-0330	9e-05	2e-12	7e-07	5e-12	8e+03	2e+05	4e-05	6e-12	2e-07	5e-12	5e+02	1e+04
1000-2-5-b-b-ns-0660	2e-04	1e-12	1e-06	9e-13	2e+04	4e+05	8e-05	5e-12	5e-07	1e-11	8e+03	2e+05
1000-2-5-b-b-ns-1000	1e-11	4e-12	1e-11	4e-12	3e+02	6e+03	2e-11	4e-12	2e-11	4e-12	6e+01	2e+03

1000-3-1-a-a-ns-0000	1e-05	3e-12	4e-07	5e-12	3e+04	7e+04	4e-06	5e-12	1e-07	7e-12	5e+02	2e+03
1000-3-1-a-a-ns-0330	8e-07	3e-12	4e-08	6e-12	7e+04	6e+05	8e-06	3e-12	4e-08	3e-12	1e+03	1e+04
1000-3-1-a-a-ns-0660	3e-06	7e-12	4e-07	9e-12	7e+04	7e+05	1e-06	1e-11	2e-07	2e-11	1e+03	1e+04
1000-3-1-a-a-ns-1000	2e-09	7e-12	2e-09	2e-12	2e+03	5e+04	3e-11	4e-12	3e-11	2e-12	1e+03	2e+04
1000-3-1-a-b-ns-0000	3e-07	2e-12	3e-09	2e-12	1e+02	1e+02	8e-08	1e-12	1e-10	3e-12	6e+01	1e+02
1000-3-1-a-b-ns-0330	2e-05	3e-12	3e-07	3e-12	4e+03	7e+04	7e-06	4e-12	3e-07	2e-12	3e+02	5e+03
1000-3-1-a-b-ns-0660	3e-06	3e-12	7e-07	4e-12	4e+03	8e+04	4e-06	3e-12	3e-07	4e-12	5e+02	1e+04
1000-3-1-a-b-ns-1000	9e-12	3e-12	9e-12	5e-12	2e+03	5e+04	4e-11	4e-12	4e-11	8e-12	3e+02	6e+03
1000-3-1-b-a-ns-0000	7e-06	5e-12	1e-06	6e-12	8e+03	5e+04	4e-09	5e-12	1e-11	3e-12	5e+02	2e+03
1000-3-1-b-a-ns-0330	2e-06	4e-12	5e-08	4e-12	2e+04	2e+05	1e-05	3e-12	7e-07	5e-12	5e+02	7e+03
1000-3-1-b-a-ns-0660	1e-06	6e-13	2e-07	1e-12	3e+04	4e+05	5e-08	1e-12	2e-08	1e-12	1e+03	1e+04
1000-3-1-b-a-ns-1000	2e-11	9e-12	2e-11	0e+00	3e+02	6e+03	4e-11	9e-12	4e-11	0e+00	6e+01	2e+03
1000-3-1-b-b-ns-0000	1e-06	6e-13	2e-07	9e-13	1e+02	4e+02	4e-08	2e-12	7e-12	2e-12	6e+01	1e+02
1000-3-1-b-b-ns-0330	1e-05	1e-12	9e-07	1e-12	2e+03	4e+04	1e-05	9e-13	6e-07	8e-13	3e+02	5e+03
1000-3-1-b-b-ns-0660	1e-05	7e-13	4e-07	2e-12	4e+03	7e+04	2e-06	6e-13	4e-08	2e-12	5e+02	9e+03
1000-3-1-b-b-ns-1000	3e-12	8e-12	3e-12	3e-12	1e+03	2e+04	1e-02	4e-12	9e-07	1e-12	6e+01	2e+03
1000-3-2-a-a-ns-0000							2e-07	2e-12	8e-13	2e-12	2e+03	3e+03
1000-3-2-a-a-ns-0330							1e-07	2e-12	4e-10	1e-12	2e+03	1e+04
1000-3-2-a-a-ns-0660	6e-06	8e-12	7e-07	8e-12	7e+04	9e+05	7e-07	8e-12	1e-08	8e-12	2e+03	3e+04
1000-3-2-a-a-ns-1000	4e-09	5e-12	4e-09	5e-13	2e+03	5e+04	1e-11	4e-12	1e-11	2e-12	1e+02	3e+03
1000-3-2-a-b-ns-0000	1e-08	2e-12	5e-10	2e-12	2e+03	3e+03	1e-05	1e-12	2e-07	2e-12	6e+01	3e+02
1000-3-2-a-b-ns-0330	3e-07	9e-13	1e-08	4e-13	4e+03	4e+04	1e-06	5e-13	3e-07	6e-13	3e+02	3e+03
1000-3-2-a-b-ns-0660	6e-07	8e-13	2e-07	8e-13	4e+03	5e+04	1e-05	5e-13	9e-08	1e-12	1e+03	1e+04
1000-3-2-a-b-ns-1000	2e-12	2e-12	2e-12	1e-12	2e+03	4e+04	2e-09	2e-12	2e-09	1e-12	3e+02	6e+03
1000-3-2-b-a-ns-0000	2e-08	1e-12	1e-09	9e-13	8e+03	8e+03	5e-07	1e-12	4e-07	2e-12	3e+02	1e+03
1000-3-2-b-a-ns-0330	1e-06	1e-12	4e-07	2e-12	8e+03	8e+04	4e-06	2e-12	4e-08	1e-12	5e+02	5e+03
1000-3-2-b-a-ns-0660	3e-06	6e-13	5e-07	2e-12	8e+03	8e+04	2e-06	2e-12	4e-08	6e-13	5e+02	6e+03
1000-3-2-b-a-ns-1000	6e-09	2e-12	6e-09	1e-12	5e+02	1e+04	9e-12	1e-11	9e-12	8e-13	3e+02	6e+03
1000-3-2-b-b-ns-0000	4e-05	2e-12	2e-07	3e-12	6e+01	2e+02	3e-07	1e-12	1e-11	3e-12	6e+01	2e+02
1000-3-2-b-b-ns-0330	3e-05	3e-12	3e-07	3e-12	4e+03	8e+04	3e-06	2e-12	3e-09	2e-12	1e+03	2e+04
1000-3-2-b-b-ns-0660	1e-05	4e-12	3e-07	5e-12	8e+03	2e+05	3e-06	3e-12	4e-09	4e-12	4e+03	9e+04
1000-3-2-b-b-ns-1000	1e-11	4e-12	1e-11	1e-11	8e+03	2e+05	5e-11	4e-12	5e-11	1e-11	2e+03	5e+04
1000-3-3-a-a-ns-0000												
1000-3-3-a-a-ns-0330												
1000-3-3-a-a-ns-0660												
1000-3-3-a-a-ns-1000												
1000-3-3-a-b-ns-0000	1e-06	8e-13	7e-07	2e-12	5e+02	2e+03	1e-06	2e-12	5e-08	3e-12	6e+01	3e+02
1000-3-3-a-b-ns-0330	2e-05	2e-12	8e-07	2e-12	4e+03	7e+04	3e-05	2e-12	5e-07	1e-12	3e+02	5e+03
1000-3-3-a-b-ns-0660	2e-06	1e-12	3e-07	2e-12	3e+04	6e+05	3e-06	5e-12	8e-07	6e-12	1e+03	2e+04
1000-3-3-a-b-ns-1000	6e-12	1e-12	6e-12	7e-13	3e+02	6e+03	2e-11	1e-12	2e-11	7e-13	6e+01	2e+03
1000-3-3-b-a-ns-0000	2e-06	2e-12	3e-08	5e-12	2e+04	3e+04	3e-05	1e-12	6e-07	2e-12	3e+02	1e+03

1000-3-3-b-a-ns-0330	3e-06	4e-12	2e-07	3e-12	2e+04	2e+05	2e-08	6e-12	3e-08	5e-12	2e+03	3e+04
1000-3-3-b-a-ns-0660	1e-05	3e-12	2e-07	3e-12	3e+04	5e+05	3e-06	2e-12	1e-08	5e-12	2e+03	3e+04
1000-3-3-b-a-ns-1000	7e-12	6e-12	7e-12	4e-13	2e+03	5e+04	1e-11	6e-12	1e-11	4e-13	1e+02	3e+03
1000-3-3-b-b-ns-0000	4e-09	3e-12	1e-13	4e-12	6e+01	7e+01	9e-10	2e-12	4e-14	2e-12	6e+01	1e+02
1000-3-3-b-b-ns-0330	2e-05	2e-12	2e-07	3e-12	8e+03	2e+05	1e-05	3e-12	3e-07	3e-12	2e+03	4e+04
1000-3-3-b-b-ns-0660	8e-06	7e-13	5e-07	4e-13	3e+04	6e+05	1e-05	9e-13	3e-07	4e-13	2e+03	4e+04
1000-3-3-b-b-ns-1000	3e-03	1e-11	9e-07	3e-12	5e+02	1e+04	9e-12	5e-12	9e-12	6e-12	3e+02	6e+03
1000-3-4-a-a-ns-0000							2e-05	2e-12	2e-07	3e-12	1e+03	3e+03
1000-3-4-a-a-ns-0330							3e-07	3e-12	9e-08	4e-12	1e+03	9e+03
1000-3-4-a-a-ns-0660							2e-08	2e-12	2e-10	2e-12	2e+03	2e+04
1000-3-4-a-a-ns-1000	5e-12	1e-11	5e-12	1e-11	5e+02	1e+04	6e-09	1e-11	6e-09	1e-11	6e+01	1e+03
1000-3-4-a-b-ns-0000	9e-07	2e-12	3e-08	2e-12	1e+02	2e+02	5e-08	2e-12	5e-10	3e-12	6e+01	3e+02
1000-3-4-a-b-ns-0330	1e-05	1e-12	1e-07	1e-12	8e+03	1e+05	2e-05	3e-12	7e-07	3e-12	3e+02	5e+03
1000-3-4-a-b-ns-0660	5e-05	2e-12	8e-07	2e-12	4e+03	8e+04	3e-05	3e-12	4e-07	3e-12	1e+03	2e+04
1000-3-4-a-b-ns-1000	5e-11	3e-12	3e-11	5e-12	5e+02	1e+04	7e-12	3e-12	6e-12	2e-12	1e+02	3e+03
1000-3-4-b-a-ns-0000	1e-09	3e-12	6e-10	6e-12	2e+04	2e+04	1e-04	3e-12	9e-07	2e-12	3e+02	2e+03
1000-3-4-b-a-ns-0330	4e-06	2e-12	6e-07	2e-12	8e+03	9e+04	6e-06	1e-12	7e-08	3e-12	5e+02	5e+03
1000-3-4-b-a-ns-0660	1e-06	3e-12	3e-08	3e-12	2e+04	2e+05	2e-07	3e-12	6e-09	4e-12	1e+03	1e+04
1000-3-4-b-a-ns-1000	4e-07	5e-12	2e-12	5e-12	5e+02	1e+04	2e-11	5e-12	2e-11	8e-12	6e+01	1e+03
1000-3-4-b-b-ns-0000	3e-07	1e-12	1e-09	2e-12	6e+01	1e+02	3e-09	2e-12	0e+00	2e-12	6e+01	1e+02
1000-3-4-b-b-ns-0330	4e-05	8e-13	7e-07	3e-12	2e+03	4e+04	3e-06	2e-12	2e-07	4e-12	1e+03	2e+04
1000-3-4-b-b-ns-0660	6e-06	2e-12	3e-07	1e-12	4e+03	8e+04	4e-06	2e-12	2e-07	3e-12	1e+03	2e+04
1000-3-4-b-b-ns-1000	2e-11	9e-12	2e-11	7e-13	4e+03	1e+05	5e-11	9e-12	5e-11	7e-13	1e+03	3e+04
1000-3-5-a-a-ns-0000	2e-06	2e-12	2e-07	3e-12	7e+04	1e+05	8e-08	2e-12	4e-12	2e-12	1e+03	2e+03
1000-3-5-a-a-ns-0330							4e-06	2e-12	2e-08	1e-12	1e+03	1e+04
1000-3-5-a-a-ns-0660							3e-05	7e-13	6e-07	2e-12	1e+03	1e+04
1000-3-5-a-a-ns-1000	2e-07	3e-12	2e-10	3e-12	3e+02	6e+03	2e-11	1e-11	2e-11	1e-11	6e+01	2e+03
1000-3-5-a-b-ns-0000												
1000-3-5-a-b-ns-0330												
1000-3-5-a-b-ns-0660												
1000-3-5-a-b-ns-1000												
1000-3-5-b-a-ns-0000	9e-09	2e-12	1e-11	2e-12	4e+03	4e+03	2e-06	2e-12	2e-09	2e-12	3e+02	1e+03
1000-3-5-b-a-ns-0330	9e-06	4e-12	4e-07	3e-12	4e+03	5e+04	3e-06	2e-12	5e-07	2e-12	3e+02	3e+03
1000-3-5-b-a-ns-0660	6e-07	2e-12	9e-07	2e-12	8e+03	1e+05	2e-07	3e-12	6e-07	2e-12	5e+02	8e+03
1000-3-5-b-a-ns-1000	8e-03	7e-12	3e-07	6e-12	5e+02	1e+04	2e-11	2e-12	2e-11	5e-12	1e+02	3e+03
1000-3-5-b-b-ns-0000	1e-08	2e-12	1e-13	3e-12	6e+01	5e+01	6e-09	5e-12	1e-15	8e-12	6e+01	1e+02
1000-3-5-b-b-ns-0330	1e-04	3e-12	5e-07	3e-12	2e+03	4e+04	1e-05	2e-12	1e-08	2e-12	5e+02	9e+03
1000-3-5-b-b-ns-0660	8e-05	3e-12	3e-07	3e-12	8e+03	2e+05	7e-05	4e-12	3e-07	4e-12	1e+03	2e+04
1000-3-5-b-b-ns-1000	1e-02	8e-12	1e-06	5e-12	3e+02	6e+03	3e-12	9e-13	2e-12	1e-11	1e+02	3e+03