

Multiple approximate-response agents (MARA): Fast near-optimal primal recovery for distributed optimization

Tetiana Parshakova

tparshakova@flatironinstitute.org

Flatiron Institute

Yicheng Bai

Amazon.com

Garrett van Ryzin

Uber AI (United States)

Stephen Boyd

Stanford University

Research Article

Keywords:

Posted Date: August 24th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10175245/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Multiple approximate-response agents (MARA): Fast near-optimal primal recovery for distributed optimization

Tetiana Parshakova*

Yicheng Bai
Stephen Boyd

Garrett van Ryzin

June 29, 2026

Abstract

Dual methods are useful for distributed optimization because they allow agent-level subproblems to be solved in parallel. However, achieving primal feasibility with dual methods is a challenge; it can take many iterations to find prices that recover primal feasibility, and even with optimal dual prices primal feasibility is not guaranteed unless special conditions like strict convexity hold. To address this limitation, we propose a simple primal recovery method, *multiple approximate-response agents* (MARA), that is able to rapidly reduce primal infeasibility, tolerating some degree of suboptimality. The method is agnostic to how dual prices are computed, so MARA can be applied to enhance any dual algorithm. Rather than returning a single primal response to each price query, MARA requires agents to generate multiple primal responses, each of which has bounded suboptimality. Because these multiple responses can be computed in parallel, there is no increase in the wall-clock time of the underlying dual algorithm. MARA then constructs a convex combination of the multiple responses by minimizing the sum of the primal and complementary slackness residuals to produce a high-quality primal solution. Tests of MARA using both a price localization method and a dual sub-gradient method show that it typically converges to a feasible, near-optimal solution in a few tens of iterations. Moreover, hyperparameters of MARA can be flexibly tuned to control the trade-off among speed, computational budget, and degree of suboptimality of the feasible solutions.

1 Introduction

Distributed optimization methods seek to decompose large optimization problems into a collection of subproblems (called *agents*) that are linked through a set of coupling constraints.

*Corresponding author. Email: tparshakova@flatironinstitute.org

Dual methods relax the coupling constraints so that the agent subproblems can be solved in parallel on distributed hardware (multiple cores and/or workers). Distributed optimization can significantly speed up the wall-clock time to find optimal solutions and greatly expand the scale of problems that can be solved. Moreover, the collection-of-agents topology of distributed optimization reflects the structure of many real-world problems: electric power systems are collections of generation, transmission and consumption agents coupled by the circuit equations; supply chains are collections of suppliers, manufacturers, distributors and retailers linked by inventory and flow balance constraints; banking operations are collections of deposit, lending and trading divisions linked by capital budget and systemic risk constraints, etc. The combination of improved solution speed, scalability and fit to real-world problems makes distributed optimization approaches attractive in practice.

However, one fundamental limitation of distributed optimization is achieving primal feasibility. Specifically, relying only on dual prices to induce primal feasibility requires high precision prices, and achieving high precision prices requires many dual iterations, potentially eroding the run time gains from parallelization. Worse, for important problem classes like linear models (more generally, problems that are not strictly convex), even optimal dual prices do not guarantee that the primal responses of agents will be feasible with respect to the coupling constraints. The problem of primal recovery is a major barrier to adoption of distributed optimization.

We propose a new method for primal recovery called *multiple approximate-response agents* (MARA) that seeks to rapidly reduce primal residuals by 1) tolerating some degree of suboptimality in agent responses, and 2) leveraging parallelization to generate multiple near-optimal primal responses from each agent. The intuition is that a large and diverse set of near-optimal agent responses can then be blended to achieve a feasible and near-optimal primal solution. More precisely, in each iteration of the dual algorithm, we ask all agents to submit multiple, approximately-optimal responses to the current dual prices instead of just one response. (Two methods to generate these multiple responses, one based on objective value suboptimality and one based on price perturbations, are given in Section 4.1.) Each response is an independent problem, so the multiple responses can be generated in parallel in the same amount of wall clock time as a single response. Next, we solve a linear program to determine a convex combination of the agents' multiple responses that minimizes the sum of primal and complementary slackness residuals, producing a blended primal solution that is closer to primal feasibility. As with the multiple responses, this auxiliary linear program can be performed as an independent parallel computation that does not impact the wall clock time of the underlying dual algorithm.

MARA can be appended to any dual algorithm, functioning as a parallel side calculation to rapidly discover feasible primal solutions while tolerating some suboptimality. Figure 1 illustrates its logical flow: The first block represents any method responsible for solving the dual problem and updating dual prices λ^k at each iteration. Regardless of the method used in this block, the second block uses the current prices λ^k and the MARA parallel calls to agents to construct a blended primal point $\bar{x}^k$ with low primal residuals.

As we show in Section 6, on a range of test problems MARA is able to discover primal

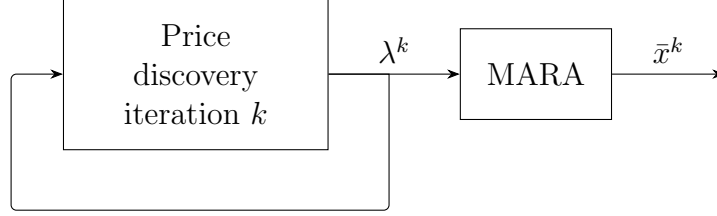


Figure 1: Price discovery method followed by MARA.

feasible, near-optimal solutions in tens of iterations, even though the primal solution from the underlying dual algorithm is still far from feasibility. Moreover, hyperparameters of MARA can be tuned to control the trade-off between number of iterations to find a near-optimal solution and the degree of suboptimality of that solution. Section 5 discusses other variations in applying MARA that can further enhance its performance. Our open source implementation will be available online.

Our open source implementation is available at:

https://github.com/cvxgrp/mra_precovery

The remainder of the paper is organized as follows. Section 2 provides a literature review, and the fundamentals of distributed optimization are discussed in Section 3. The components of MARA, including the two multiple-response oracles, are defined in Section 4 and extensions and variations of MARA are presented in Section 5. Section 6 describes our numerical tests of MARA and conclusions are given in Section 7.

2 Literature review

The issue of primal points not converging to feasibility in dual methods has long been acknowledged [Sho12, LL97]. The traditional and simple solution is to perform primal averaging to recover primal variables in the limit [LPS99]. For example, [SJR16] use primal averaging in a consensus-based dual decomposition framework to obtain approximate primal solutions. Other works explored more sophisticated strategies. For example, [SS86] combine a projected subgradient method with an auxiliary penalty approach to produce primal-dual iterates. And [CJOS24] propose a recursive regularization scheme that requires an approximate primal regularized dual solve and a query to a non-separable dual interface over the primal feasibility set in each round under a differentiable Lagrangian assumption. None of these methods, however, use the same oracles and problem assumptions as MARA (see §3) and, therefore, cannot be applied directly.

Wang [Wan17] develops a related coordination method for large nonconvex multi-agent problems. The method updates dual prices and selects one best response per agent. Using the Shapley-Folkman theorem, it exploits the fact that the sum of many nonconvex response sets is nearly convex, yielding an asymptotically vanishing price of decentralization. MARA

uses a related but distinct convexification step: at a fixed, possibly suboptimal price, each convex agent supplies multiple approximate responses, and MARA optimizes over their convex hull to reduce primal and complementary slackness residuals. Thus, Wang combines price discovery with single-response selection for nonconvex agents, whereas MARA retains convex mixtures and is agnostic to the price-discovery method.

Subgradient methods for solving dual problems have been widely used in network applications to develop distributed methods [LL99]. A key concern is to recover near-feasible and near-optimal primal solutions using dual subgradients. Primal averaging tackles this by constructing an ergodic sequence of primal variables that converges to a primal solution asymptotically [LPS99]. It was first proposed in the context of primal-dual subgradient schemes [NJ78] and linear problems [Sho85]. In [NO09], the authors showed that primal averaging in dual subgradient methods with constant step size (as opposed to commonly used diminishing step size) recovers approximate primal optimal solutions. Even though we are unaware of results applying primal averaging to dual localization methods to recover primal solutions, we compare our approach to primal averaging in the numerical examples.

There have been decades of research on optimization methods built around neutral cutting-plane oracles. For example, subgradient methods were developed by Shor in 1962 [Sho12]. Extensions such as projected subgradient, Polyak’s step, primal-dual subgradient methods, dual averaging, and heavy ball methods have further broadened their scope; see [CFM75, Pol87, BXM03, Nes13]. Subgradient methods are first-order methods; thus, their convergence is often slow and depends on the conditioning of the problem. However, their low per-iteration cost makes subgradient methods suitable for large-scale problems.

Another line of work that uses subgradient oracles involves localization methods [BV07], *i.e.*, cutting-plane and ellipsoid methods. These methods require more computation per iteration but typically are much faster than subgradient methods. Many variations have been proposed, including analytic center cutting-plane method [Son88, GV93], maximum volume ellipsoid cutting-plane method [Tar88], volumetric center cutting-plane method [Vai89], and Chebyshev center cutting-plane method [EM75]. Using any of these methods, we can obtain optimal dual prices conforming to our access assumptions (see §3.1). Interestingly, there have been attempts to marry the two ideas of subgradient methods with ellipsoid methods [RN23], but achieving fast practical convergence in such hybrid schemes remains an open research question.

In our numerical experiments, we use the homogeneous analytic center cutting-plane method (ACCPM) [NV99]. For more details and an overview of the related methods, see [Pét02, §6,7]. A cutting-plane approach typically leads to faster convergence. This is especially beneficial when agent queries are costly, since it allows us to quickly obtain a good estimate of the dual variable. Specifically, we choose ACCPM because it works well in practice and allows the approximate analytic center to be efficiently computed using the Newton method of infeasible start with diagonal Hessian [BVS08].

3 Distributed optimization

Consider the optimization problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && Ax \leq b, \end{aligned} \tag{1}$$

with variable $x \in \mathbf{R}^n$, where $f : \mathbf{R}^n \rightarrow \mathbf{R} \cup \{\infty\}$ is convex, closed and proper, $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given, and the inequality is entry-wise.¹ In distributed optimization, we are interested in the special case of (1) when f is block separable,

$$f(x) = \sum_{i=1}^K f_i(x_i),$$

where $x = (x_1, \dots, x_K)$, with $f_i : \mathbf{R}^{n_i} \rightarrow \mathbf{R} \cup \{\infty\}$ convex, closed and proper. We refer to f_i as the objective function of agent i . Breaking the matrix A into blocks conformable with $(x_1, \dots, x_K)$, we can then express the problem (1) as

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^K f_i(x_i) \\ & \text{subject to} && \sum_{i=1}^K A_i x_i \leq b. \end{aligned} \tag{2}$$

We refer to the separate blocks i as *agents*. This block-separable form makes the use of distributed optimization via dual methods appealing.

3.1 Price-directed oracle

We assume that $\text{dom } f$ is bounded and that for any $y \in \mathbf{R}^n$ we can compute

$$x(y) \in \underset{z \in \text{dom } f}{\text{argmin}} (f(z) - y^T z). \tag{3}$$

We refer to this oracle as *price-directed*, since it returns an optimal action $x(y)$ given prices y . This oracle is typically the primal step in a primal-dual algorithm. When f is strictly convex, the argmin is unique. However, when strict convexity does not hold, there may be multiple optimal responses $x(y)$, in which case we assume that the oracle (3) returns one of the optimal responses. This indeterminacy in the oracle's response makes primal recovery challenging. For notational convenience, we often omit writing the y -dependence in $x(y)$.

The price-directed oracle is closely related to f^* , the conjugate of f , as

$$f^*(y) = \sup_{x \in \text{dom } f} (y^T x - f(x)) = y^T x(y) - f(x(y)).$$

¹Note that affine equality constraints can be formulated as a pair of opposing affine inequalities, and therefore handled in the problem (1). Moreover, while we focus on affine constraints here for ease of exposition, more general convex constraints can be handled using MARA.

We have $x(y) \in \partial f^*(y)$, where ∂ denotes subdifferential. So, the oracle (3) is the same as finding a subgradient of the conjugate of f at y .

Using the block separable form of f , we can split (3) into price-directed oracles for each f_i , *i.e.*,

$$x_i(y_i) \in \underset{z_i \in \text{dom } f_i}{\text{argmin}} (f_i(z_i) - y_i^T z_i), \quad i = 1, \dots, K, \quad (4)$$

where $y = (y_1, \dots, y_K)$, conformable with x . This property enables parallelization of the price-directed oracles.

3.2 Optimality conditions

Assuming that there exists an x in the relative interior of $\text{dom } f$ that satisfies $Ax \leq b$, then since the problem (1) has only affine constraints, strong duality holds from Slater's condition [BV04, §5.2.3]. Therefore, the Karush-Kuhn-Tucker (KKT) optimality conditions for (1) are

$$x = x(y), \quad y = -A^T \lambda, \quad (5)$$

$$Ax \leq b, \quad (6)$$

$$\lambda_i (Ax - b)_i = 0, \quad i = 1, \dots, m, \quad (7)$$

$$\lambda \geq 0, \quad (8)$$

where $\lambda \in \mathbf{R}^m$ is the dual variable (Lagrange multiplier) associated with the inequality constraints $Ax \leq b$. Assuming $\lambda \geq 0$, $y = -A^T \lambda$, and $x = x(y)$, stationarity (5) and dual feasibility (8) are automatically satisfied. Thus, the optimality conditions are reduced to (6) and (7), which are the primal feasibility and the complementary slackness, respectively.

The dual variable λ has the interpretation of the price vector for violating the inequalities (6), and the variable $y = -A^T \lambda$ has the interpretation of the price vector local to the primal variable x . Note that y can be decomposed into price vectors $y_i = -A_i^T \lambda$ local to each agent's primal variable x_i . Define the optimal primal variable x^* and the optimal prices λ^* . When f is strictly convex, x^* is unique, but the price vector λ^* need not be.

Suppose we are given λ^* and $\tilde{x}$ satisfying (5). Then $\tilde{x}$ need not satisfy (6) and (7) unless special conditions hold, *e.g.*, f is strictly convex. In this case, x^* is unique and therefore $\tilde{x}$ must be the optimal point. However, strict convexity rules out many practically relevant cases, such as linear models. Moreover, even in the strictly convex case, convergence can be slow because recovering a primal solution from the dual alone often requires λ^* to be known with high precision, and achieving such high precision can require a large number of dual iterations. This behavior makes primal recovery a challenge in distributed optimization.

3.3 Optimality condition residuals

We define the primal and complementary slackness residuals for a pair $(x, \lambda) \in \mathbf{R}^n \times \mathbf{R}_+^m$ as

$$r_p = \mathbf{1}^T (Ax - b)_+, \quad r_c = \lambda^T |Ax - b|,$$

where $\mathbf{1}$ is the vector with all entries equal to one, $u_+ = \max\{u, 0\}$ is the positive part, and the positive part and absolute values are applied entrywise. Evidently r_p and r_c are nonnegative, with (x, λ) optimal (assuming $\lambda \geq 0$, $y = -A^T \lambda$, and $x = x(y)$) if and only if $r_p + r_c = 0$. We use these residuals as a stopping criterion,

$$r_p + r_c \leq \epsilon_r,$$

where $\epsilon_r > 0$ is a given residual tolerance.

3.4 Dual subgradients

The dual function associated with the problem (1) is

$$g(\lambda) = -f^*(-A^T \lambda) - \lambda^T b$$

for $\lambda \geq 0$. The optimal prices λ^* maximize g over $\lambda \geq 0$. The negative dual function $-g$ is convex, with subgradient at $\lambda \geq 0$

$$-Ax(y) + b \in \partial(-g)(\lambda), \tag{9}$$

which we can construct from our agent price-directed oracles. These subgradients are used in various primal-dual algorithms.

4 Components of MARA

We next describe the two main components of MARA: 1) oracles for generating multiple responses, and 2) a primal recovery method that forms a primal solution by blending the multiple responses.

4.1 Multiple approximate-response oracles

For any $y \in \mathbf{R}^n$, an *approximate price-directed oracle* returns $x^{\text{apx}}(y)$ that approximately minimizes $f(z) - y^T z$ with respect to z , i.e.,

$$-f^*(y) \leq f(x^{\text{apx}}(y)) - y^T x^{\text{apx}}(y) \approx -f^*(y). \tag{10}$$

We next construct two approximate oracles that ensure that any convex combination of ϵ -suboptimal points results in only a small violation of the stationarity condition (5).

4.1.1 Value suboptimality oracle

For any $y \in \mathbf{R}^n$, we define an ϵ_v -value suboptimal primal variable with respect to (10) as $x^v(y)$ such that

$$-f^*(y) \leq f(x^v(y)) - y^T x^v(y) \leq -f^*(y) + \epsilon_v |f^*(y)|.$$

To implement this oracle, we follow the approach described in [SB10]. Specifically, we sample N i.i.d. directions $\delta_1, \dots, \delta_N \sim \mathcal{N}(0, I_n)$. For each $j = 1, \dots, N$, we compute z_j by solving a convex problem

$$\begin{aligned} & \text{maximize} && \delta_j^T z_j \\ & \text{subject to} && f(z_j) - y^T z_j \leq -f^*(y) + \epsilon_v |f^*(y)|. \end{aligned}$$

The solutions $\{z_1, \dots, z_N\}$ form a list of N ϵ_v -suboptimal primal variables associated with local price y . Further for any convex combination of these points, $\bar{x} = \sum_{j=1}^N \theta_j z_j$ with $\mathbf{1}^T \theta = 1$ and $\theta \geq 0$, the suboptimality w.r.t. stationarity condition (5) is preserved

$$f(\bar{x}) - y^T \bar{x} \leq \sum_{j=1}^N \theta_j (f(z_j) - y^T z_j) \leq -f^*(y) + \epsilon_v |f^*(y)|.$$

Note this oracle requires a modification of the agent subproblems.

4.2 Price perturbation oracle

For any $y \in \mathbf{R}^n$, we also define the primal point $x^p(y)$ with ϵ_p -perturbed prices with respect to (10) as

$$f(x^p(y)) - (y + \delta)^T x^p(y) = -f^*(y + \delta),$$

with perturbation vector $\delta \in [-\epsilon_p |y|, \epsilon_p |y|]$. To compute this, we sample $\delta_1, \dots, \delta_N$ uniformly at random from the bounding box $[-\epsilon_p |y|, \epsilon_p |y|]$. For each $j = 1, \dots, N$, we compute $z_j \in \partial f^*(y + \delta_j)$. The solutions $\{z_1, \dots, z_N\}$ form a list of N ϵ_p -perturbed prices with respect to y .

If we assume that f^* is L -smooth, we can get the bound on the suboptimality of this oracle. Recall that f^* is L -smooth if and only if f is $(1/L)$ -strongly convex. Then for any convex combination $\bar{x} = \sum_{j=1}^N \theta_j z_j$ we have

$$\begin{aligned} f(\bar{x}) - y^T \bar{x} &\leq \sum_{j=1}^N \theta_j (f(z_j) - y^T z_j) \\ &\leq \sum_{j=1}^N \theta_j (f(x) - (y + \delta_j)^T (x - z_j) - y^T z_j) \\ &= -f^*(y) + \sum_{j=1}^N \theta_j \|\delta_j\|_2 \|x - z_j\|_2 \\ &\leq -f^*(y) + \epsilon_p^2 L \|y\|_2^2. \end{aligned}$$

In the above, we used Jensen's inequality for convex f , subgradient inequality $f(z_j) \leq f(x) - (y + \delta_j)^T (x - z_j)$, Cauchy-Schwarz inequality, $\|\delta_j\|_2 \leq \epsilon_p \|y\|_2$. Also by L -smoothness of f^* , we have $\|x - z_j\|_2 = \|\nabla f^*(y) - \nabla f^*(y + \delta_j)\|_2 \leq L \|\delta_j\|_2$. The point $\bar{x}$ is suboptimal

with an additive error that is proportional to the square of the price perturbation's norm. Note that because it is price based, this oracle does not require any modification to the agent subproblems.

4.3 Primal recovery method

We next define a primal recovery method that uses the multiple agent responses to generate a new primal variable which reduces the sum of residuals, $r_p + r_c$. This recovery procedure is an independent post-processing step that can be applied flexibly, either at every iteration of the dual algorithm, applied periodically (*e.g.*, every 10 iterations), or only applied after some measure of dual prices have stabilized.

Starting from a given dual variable $\lambda \in \mathbf{R}_+^m$, it consists of two steps:

Step 1. Approximate oracle queries Using an approximate oracle from §4.1, each agent $i = 1, \dots, K$ returns a list of N_i ϵ -suboptimal primal vectors, $\{z_{i_1}, \dots, z_{i_{N_i}}\}$, associated with local price vector $y_i = -A_i^T \lambda$. Note that this step can be computed by making parallel calls to agents, so the wall-clock time is equivalent to a single agent response.

Step 2. Primal recovery Construct a convex combination of the responses of each agent, $\bar{x}_i$, minimizing the sum of residuals, $r_p + r_c$, by solving the following linear programming problem:

$$\begin{aligned} & \text{minimize} && r_p + r_c \\ & \text{subject to} && r_p = \mathbf{1}^T (A\bar{x} - b)_+ \\ & && r_c = \lambda^T |A\bar{x} - b| \\ & && \bar{x}_i = Z_i u_i, \quad i = 1, \dots, K \\ & && \mathbf{1}^T u_i = 1, \quad i = 1, \dots, K \\ & && u_i \geq 0, \quad i = 1, \dots, K \\ & && \bar{x} = (\bar{x}_1, \dots, \bar{x}_K) \end{aligned} \tag{11}$$

with variables $u_i \in \mathbf{R}^{N_i}$ for all $i = 1, \dots, K$, and matrices

$$Z_i = \begin{bmatrix} z_{i_1} & \cdots & z_{i_{N_i}} \end{bmatrix} \in \mathbf{R}^{n \times N_i}, \quad i = 1, \dots, K. \tag{12}$$

Using results in §4.1, the primal point $\bar{x}$ introduces small violations of the stationarity condition (5) while simultaneously minimizing the sum of residuals, $r_p + r_c$.

Remark 1. Note that since $\bar{x}$ is constructed from primal points obtained using an approximate oracle, $r_p + r_c = 0$ does not guarantee that $\bar{x}$ is primal optimal. This is because $\bar{x}$ need not respect the stationarity condition (5).

5 Extensions and variations

There is significant flexibility in applying MARA. In this section, we highlight a few key variations of the method.

5.1 Minimizing r_p

First, since we are ultimately interested in reducing primal residuals, problem (11) can be replaced by the one below

$$\begin{aligned}
& \text{minimize} && r_p \\
& \text{subject to} && r_p = \mathbf{1}^T (A\bar{x} - b)_+ \\
& && \bar{x}_i = Z_i u_i, \quad i = 1, \dots, K \\
& && \mathbf{1}^T u_i = 1, \quad i = 1, \dots, K \\
& && u_i \geq 0, \quad i = 1, \dots, K \\
& && \bar{x} = (\bar{x}_1, \dots, \bar{x}_K),
\end{aligned} \tag{13}$$

where we removed the complementary slackness residual r_c from the objective in (11). Originally, the complementary slackness residual r_c was included in the objective to prevent us from generating a primal point that is too suboptimal. However, due to the convexity of f_i for each $i = 1, \dots, K$, if each candidate point in Z_i has tolerable suboptimality with respect to $f_i(x_i^*)$, then so does any convex combination. Consequently, the problem (13) will find a point $\bar{x}$ that also has tolerable suboptimality, with smaller primal residuals than the point returned by (11).

5.2 Increasing the primal set with history

If the approximate oracle generates primal points with comparable function values in the last H_i iterations, then we can add these additional candidate points, $Z_i^{k-H_i+1}, \dots, Z_i^k$ for all $i = 1, \dots, K$, into the candidate set of responses used to construct a new point $\bar{x}^k$ using (13). Due to convexity of the agents' functions, the resulting point will have a comparable function value, while searching over the larger polytope to reduce the primal residuals.

5.3 Oracle suboptimality mixing

The suboptimality in the approximate oracle controls the violation of the stationarity condition (5). In particular, let $\bar{x}_1$ and $\bar{x}_2$ be returned by MARA primal recovery with additive suboptimality $\epsilon_1 \gg \epsilon_2$, respectively, *i.e.*,

$$-f^*(y) \leq f(\bar{x}_j) - y^T \bar{x}_j \leq -f^*(y) + \epsilon_j, \quad j = 1, 2.$$

This implies

$$f(\bar{x}_1) \gg f(\bar{x}_2) - \|y\|_2 \|\bar{x}_2 - \bar{x}_1\|_2.$$

Therefore, if $\|y\|_2 \|\bar{x}_2 - \bar{x}_1\|_2$ is sufficiently small, the primal point $\bar{x}_2$ will have a lower function value (*i.e.*, $f(\bar{x}_1) > f(\bar{x}_2)$), but potentially higher residuals $r_p + r_c$. This is because the set of ϵ_2 -suboptimal primal points is a subset of the ϵ_1 -suboptimal points. This suggests splitting the budget of responses N_i across multiple approximate oracles with varying levels of suboptimality. This approach balances between suboptimality and primal feasibility, allowing us to quickly obtain the primal feasible point with large ϵ , while still allowing further improvement in suboptimality and feasibility as the dual iterate improves.

6 Numerical results

To test MARA we ran numerical experiments using two different dual methods applied to four problem classes. We first discuss the two dual methods, then describe the experimental process and finally discuss the results of MARA on each of the four problem classes.

6.1 Price discovery methods

6.1.1 Projected dual subgradient method

Consider first a subgradient method to solve the dual problem [BXM03]. Let $\mathcal{P}$ be the initial set containing optimal dual variable, *e.g.*, $\mathcal{P} = \mathbf{R}_+^m$. Let $\Pi_{\mathcal{P}} : \mathbf{R}^m \rightarrow \mathcal{P}$ denote the Euclidean projection onto this set. Then the projected dual subgradient method is given by Algorithm 6.1. Note that this algorithm requires the feasible set $\mathcal{P}$, initial point λ^0 and the step size rule to update α^k .

Algorithm 6.1 PROJECTED DUAL SUBGRADIENT METHOD

given feasible set $\mathcal{P}$, initial point λ^0 .
for iteration $k = 0, 1, \dots$
 Query (9) at λ^k to get a $g(\lambda^k)$ and $q^k \in \partial(-g)(\lambda^k)$.
 Set the step size α^k .
 Update iterate $\lambda^{k+1} = \Pi_{\mathcal{P}}(\lambda^k - \alpha^k q^k)$.
 Compute best value $g_{\text{best}}^k = \max\{g(\lambda^k), g_{\text{best}}^{k-1}\}$.
Check stopping criterion. Quit if $g_{\text{best}}^k - g_{\text{best}}^{k-1}$ is small.

6.1.2 Price localization method

Unlike dual subgradient methods, which rely on simple steps based on excess consumption, localization methods perform substantial calculation to update dual prices at each iteration. As a result, they tend to produce higher-quality dual prices within a small number of iterations. Moreover, localization methods do not rely on function evaluations, which enables them to handle more general cases when the agents do not have an objective and instead use a monotone operator that maps prices to actions.

Recall (9), then subgradient inequality for the convex function $-g$, together with $g(\lambda^*) \geq g(\lambda)$, gives

$$(-Ax(y) + b)^T(\lambda^* - \lambda) \leq 0,$$

i.e., any λ^* lies in the halfspace

$$\left\{ \tilde{\lambda} \mid \underbrace{(-Ax(y) + b)^T}_{=:c} \tilde{\lambda} \leq \underbrace{(-Ax(y) + b)^T \lambda}_{=:d}, y = -A^T \lambda \right\} = \left\{ \tilde{\lambda} \mid c^T \tilde{\lambda} \leq d \right\}. \quad (14)$$

Thus by evaluating $x(y)$ for $\lambda \geq 0$ (which is consistent with our access assumptions) we obtain a cutting plane for λ^* that passes through λ , *i.e.*, a neutral cutting plane for the optimal price.

Localization methods for solving (1) use the neutral cutting plane to refine the set that contains the optimal price λ^* . They differ in how they find the “center” point of the current set from which to query the cutting plane. We use the homogeneous analytic center cutting-plane method [NV99], which embeds the problem into extended space, normalizes the iterates via the proximal term, and offers more flexibility in terms of initial information than related methods. Specifically, this involves embedding the dual problem into a conic form with augmented variable $z = (t, \bar{\lambda}) \in \mathbf{R}_+ \times \mathbf{R}^m$ such that the dual variable is $\lambda = \bar{\lambda}/t$. This method also employs augmented self-concordant barriers, given by the sum of a self-concordant barrier for the cone and a proximal term. To summarize, the localization method is described in Algorithm 6.2.

Algorithm 6.2 HOMOGENEOUS ANALYTIC CENTER CUTTING-PLANE METHOD

given initial ν -normal barrier $F(z)$.

Set initial function $F^0(z) = F(z) + (1/2)\|z\|_2^2$.

for iteration $k = 0, 1, \dots$

Compute center point $(t^{k+1}, \bar{\lambda}^{k+1}) = \operatorname{argmin}_z F^k(z)$.

Query (14) at $\lambda^{k+1} = \bar{\lambda}^{k+1}/t^{k+1}$ to get a halfspace $\{\lambda \mid (c^{k+1})^T \lambda \leq d^{k+1}\}$.

Compute normalization $n^{k+1} = (\|c^{k+1}\|_2^2 + (d^{k+1})^2)^{1/2}$.

Update the function $F^{k+1}(z) = F^k(z) - \log((td^{k+1} - (c^{k+1})^T \bar{\lambda})/n^{k+1})$, where $z = (t, \bar{\lambda})$.

Check stopping criterion. Quit if $\|\lambda^{k+1} - \lambda^k\|_2$ is small.

For the initial barrier $F(z)$ we use the logarithmic barrier

$$F(z) = - \sum_{j=1}^l \log(td^j - (c^j)^T \bar{\lambda}) - \log t,$$

which is $(l+1)$ -normal barrier function. Algorithm 6.2 converges to ϵ accuracy in $O(\nu/\epsilon^2)$ iterations. For more details see [Pét02, §6,7] and [NV99]. In localization methods, *e.g.*, ACCPM, we establish the convergence of the best-to-date function value. Interestingly, in homogeneous ACCPM, we can construct a sequence of points that ensures the convergence of the function values; see Appendix A.

6.2 Numerical testing procedure

We used the following procedure to test MARA: For every problem class, at every iteration k the given price discovery method computes a dual vector λ^k using agent price-directed oracles $x^k \in \partial f^*(-A^T \lambda^k)$. We assume we have a good approximation of the range of optimal prices, thus we set the initial polyhedron for the localization method within a box that spans from

1/3 to 3 times the estimated maximum and minimum price values. We also use this box as the feasible set $\mathcal{P}$ in the dual subgradient method. For the step size rule, we select the value from the set $\alpha^k \in \{0.1/\sqrt{k}, 1/\sqrt{k}, 1/k, 10/k\}$ that minimizes the final primal residuals for each example.

We then applied the MARA primal recovery method to compute $\bar{x}^k$ with respect to prices $y^k = -A^T \lambda^k$ in each iteration. We tried both the value suboptimality and price perturbation approximate oracles for each example. We also report the suboptimality of the function value for the projection of the primal point x^k onto the primal feasibility set

$$\Pi(x^k) = \operatorname{argmin}_{x: Ax \leq b} \|x - x^k\|_2.$$

Lastly, since primal averaging in dual subgradient methods is a common technique to recover the primal variable in the limit [LPS99], for comparison we also report the performance of the primal average point, $\frac{1}{k} \sum_{j=1}^k x^j$, as another benchmark.

We consider the solution primal feasible if the relative primal infeasibility satisfies $\|r_p\|/\|b\|_2 < 10^{-6}$. Results for every problem class are reported in the same four panel figure (*e.g.*, see Figure 2), where for every experiment we plot best-to-date suboptimality of the latest feasible point in the top left panel; any improvement is marked with a star $\star$. If a method never recovers a feasible point, its curve will not appear in this panel.

We initially set the suboptimality threshold for approximate response to $\epsilon = 10\%$ and number of approximate responses $N_i = 10$ for all $i = 1, \dots, K$. In §6.7 we also compare the performance of MARA for $\epsilon = 10\%$ and $\epsilon = 1\%$ as well as $N_i = 10$ and $N_i = 50$ for all $i = 1, \dots, K$. Lastly, in §6.8 we discuss the performance of MARA when history is included, *i.e.*, $H \geq 2$.

6.3 Resource allocation problem

As a first example we consider the resource allocation problem,

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^K f_i(x_i) \\ & \text{subject to} && \sum_{i=1}^K x_i \leq R, \end{aligned}$$

where $f_i : \mathbf{R}_+^m \rightarrow \mathbf{R} \cup \{\infty\}$ is a negative utility function, $R \in \mathbf{R}_+^m$ is a vector with budgets for m resources, and $x_i \in \mathbf{R}_+^m$ are variables for all $i = 1, \dots, K$. Our example uses negative utility functions of the form

$$f_i(x_i) = \begin{cases} -\mathcal{G}(C_i x_i) & x_i \geq 0, \ x_i^T \mathbf{1} \leq 1 \\ \infty & \text{otherwise,} \end{cases}$$

where $\mathcal{G}(u) = (\prod_{i=1}^p u_i)^{1/p}$ for $u \in \mathbf{R}_+^p$ is the geometric mean function. The entries of C_i are nonnegative, so $-f_i$ is concave and nondecreasing.

Problem instance We choose the entries of the j -th column of C_i to be uniform on $[(j-1)/m, j/m]$, so higher-indexed resources are more efficient. The resource budgets form a decreasing sequence $R_j = \sqrt{K} + (K/2 - \sqrt{K})(m-j)/(m-1)$ for $j = 1, \dots, m$, so more efficient resources are also more scarce. We consider $m = 50$ resources and $K = 100$ agents. The utility functions use $p = 5$, *i.e.*, each is the geometric mean of p affine functions. The dimension of primal variable is 5000 and of dual variable is 50.

Results Figure 2 illustrates the suboptimality and relative residuals achieved with the localization method for the resource allocation problem. Note that in each iteration MARA using the value suboptimality oracle with $\epsilon_v = 10\%$ recovers a feasible primal point $\bar{x}^k$ (relative primal residuals in the lower left panel are in the $[10^{-11}, 10^{-9}]$ range), achieving, for example, 1.2% function value suboptimality in iteration 25. At iteration 92, MARA using a price perturbation oracle with $\epsilon_p = 10\%$ finds a feasible primal point $\bar{x}^k$ with a function value suboptimality of 0.1%.

In contrast, at iteration 95, the original primal value x^k is still primal infeasible with 16.7% relative infeasibility and 3.2% suboptimality, and even the primal average point at $k = 99$ remains infeasible with 7.2% relative infeasibility and 1.7% suboptimality. Moreover, the projected point $\Pi(x^k)$ is everywhere outside the $\text{dom } f$ and therefore has unbounded negative utility.

Similarly, Figure 3 illustrates the suboptimality and relative residuals achieved with the dual subgradient method. MARA using a value suboptimality oracle with $\epsilon_v = 10\%$ produces a primal feasible point at the first iteration with 0.7% suboptimality of the function value. However, MARA using a price perturbation oracle with $\epsilon_p = 10\%$ failed to find a feasible solution within 100 iterations. The competing methods never recover a primal feasible point within 100 iterations, with relative infeasibility ranging from 53% to 90% and suboptimality ranging from 24% to 34%.

6.4 Assignment problem

In this example, we solve the convex relaxation of the nonconvex problem. Specifically, we examine the assignment problem, which involves assigning different projects to teams on the same network in a way that minimizes the total cost while respecting the capacity constraints of the teams. We consider a network that is a complete bipartite graph with each edge (i, j) connecting one of the projects $i = 1, \dots, n$ to each of the teams $j = 1, \dots, m$. We let individual projects and teams be agents, *i.e.*, the total number of agents is $K = n + m$. Then the distributed optimization problem is given by

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^n \pi_i(x_i) + \sum_{j=1}^m \tau_j(c_j) \\ & \text{subject to} && \sum_{i=1}^n (x_i)_j \leq c_j, \quad j = 1, \dots, m, \end{aligned}$$

where $\pi_i : \mathbf{R}_+^m \rightarrow \mathbf{R} \cup \{\infty\}$ is the objective of the project agent i , $\tau_j : \mathbf{R}_+ \rightarrow \mathbf{R} \cup \{\infty\}$ is the objective of the team agent j , and $x_i \in \mathbf{R}_+^m$, $c_j > 0$ are variables for all $i = 1, \dots, n$ and $j = 1, \dots, m$.

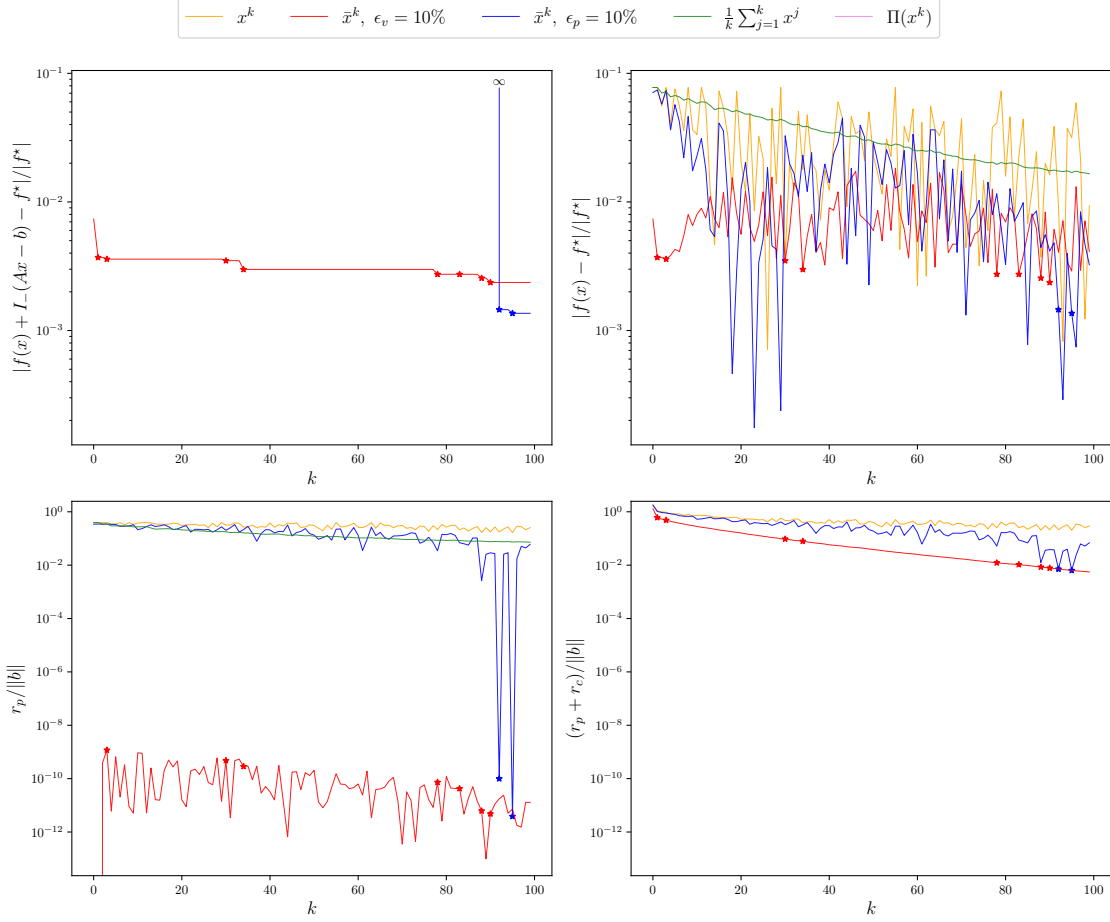


Figure 2: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the resource allocation problem.

We define $\pi_i(x_i)$ as the optimal value to the problem

$$\begin{aligned}
 & \text{minimize} && -r_i^T \tilde{x}_i \\
 & \text{subject to} && q_i \tilde{x}_i \leq x_i \\
 & && \tilde{x}_i^T \mathbf{1} \leq 1 \\
 & && \tilde{x}_i \in \{0, 1\}^m,
 \end{aligned} \tag{15}$$

where $\tilde{x}_i$ is a variable, $r_i \in \mathbf{R}_+^m$ is a vector with rewards for assigning project i to each of the m teams, and $q_i > 0$ is the consumed capacity for assigning project i . The last two constraints correspond to assigning a single project to at most one team.

We define $\tau_j(c_j)$ as the optimal value to the problem

$$\begin{aligned}
 & \text{minimize} && a_j(\tilde{c}_j - d_j)_+^2 \\
 & \text{subject to} && c_j \leq \tilde{c}_j, \quad \tilde{c}_j \geq 0
 \end{aligned}$$

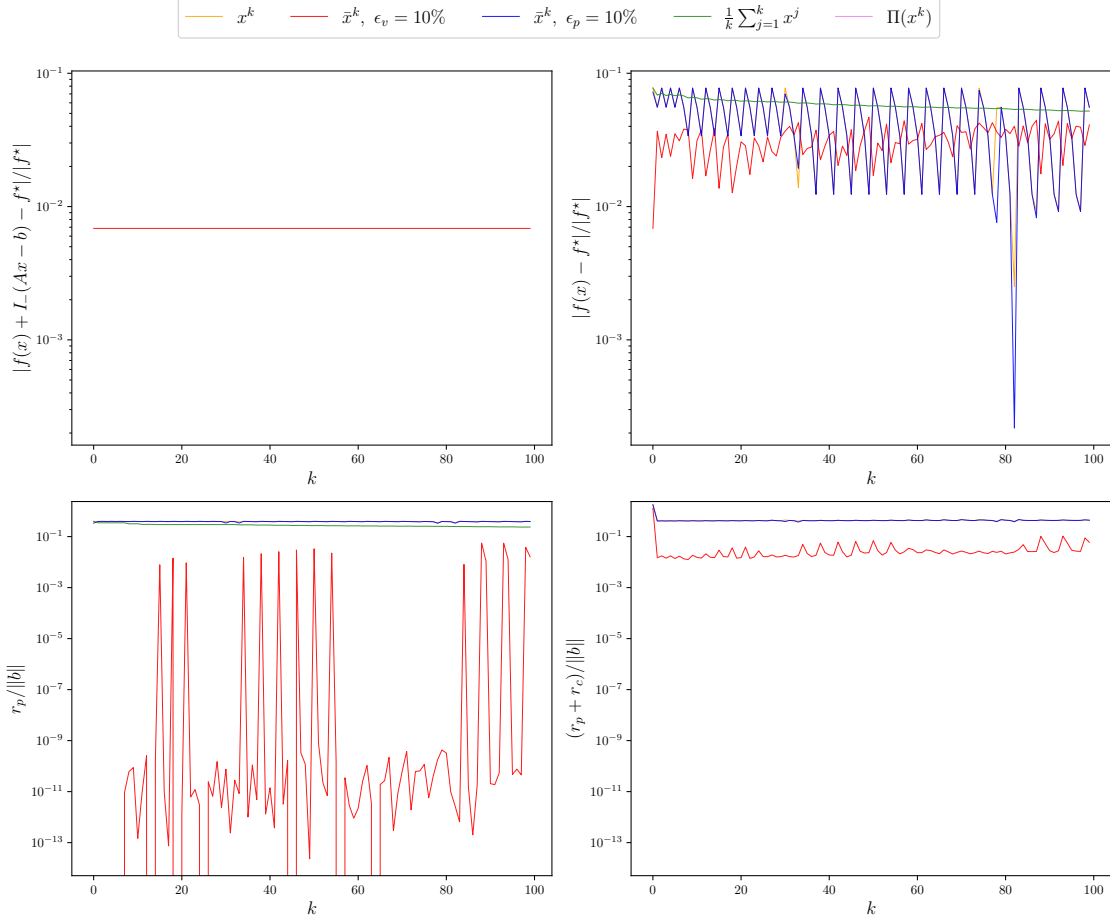


Figure 3: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus dual subgradient method iterations for the resource allocation problem.

where $\tilde{c}_j$ is a variable, $a_j, d_j > 0$. Thus $\tau_j(c_j)$ is an increasing convex function.

Problem instance We choose the entries of r_i to be uniform in $[0, 1]$. The values q_i are sampled randomly from $\{1, \dots, \lceil n/2m \rceil\}$. The values a_j are sampled uniformly from $[0.8w, 1.25w]$ for $w = (\sum_{i=1}^n \mathbf{1}^T r_i) / (30q^T \mathbf{1})$.

We sample the values d_j randomly from $\{1, \dots, \lceil q^T \mathbf{1} / n \rceil\}$. For this specific instance, we consider $n = 200$ projects and $m = 50$ teams. The dimension of primal variable is 10050 and the dual variable has dimension 50.

To obtain convex relaxation of the assignment problem, we replace the integer constraints in (15) with the box constraints $\tilde{x}_i \in [0, 1]^m$ for all $i = 1, \dots, n$. This corresponds to taking the convex closure of the objective function in assignment problem. Since the problem is convex, we proceed with a convex version of the primal recovery method.

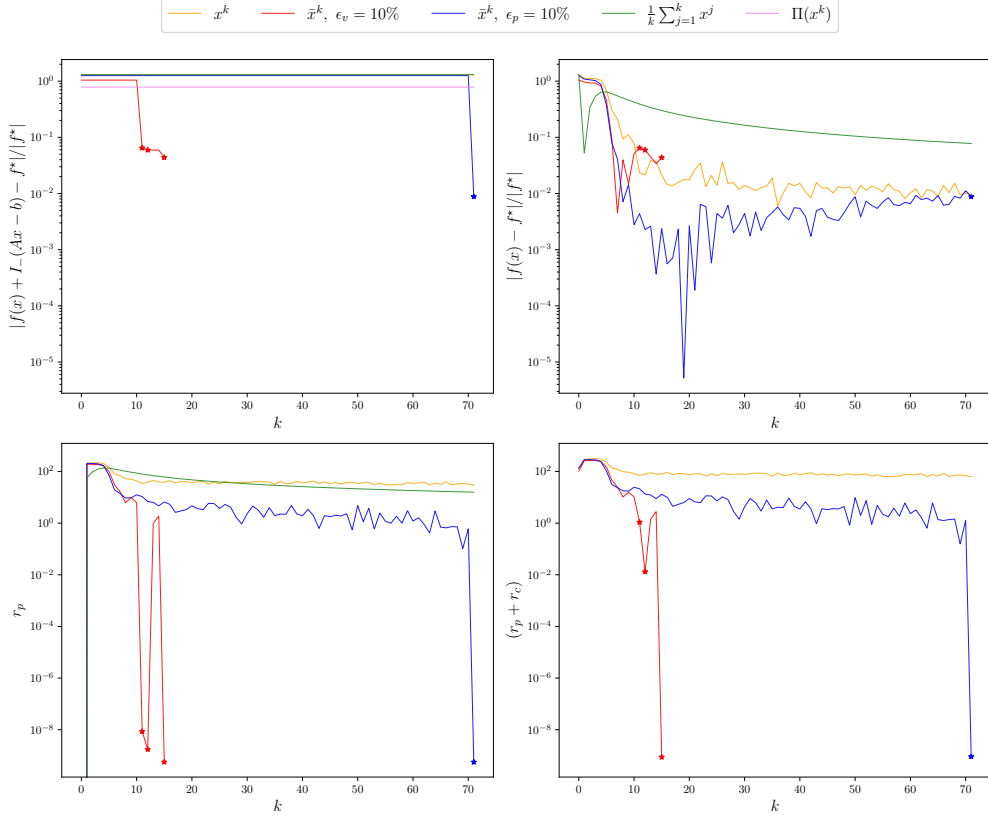


Figure 4: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), primal violations (bottom left), residuals (bottom right) versus localization method iterations for convex relaxation of the assignment problem.

Results Figure 4 illustrates the suboptimality and relative residuals achieved with the localization method for the assignment problem. At iteration 15, the $\epsilon_v = 10\%$ primal recovery method finds a feasible primal point $\bar{x}^k$ with 4.4% suboptimality of the function value. At iteration 71, the $\epsilon_p = 10\%$ primal recovery method finds a feasible primal point $\bar{x}^k$ with a function value suboptimality of 0.9%. In contrast, the primal points x^k , primal average points $\frac{1}{k} \sum_{j=1}^k x^j$, and the projected points $\Pi(x^k)$ are primal feasible only at the initial iteration, with suboptimality of 130%, 130% and 79%, respectively.

Similarly, Figure 5 illustrates suboptimality and relative residuals achieved with the dual subgradient method. Although all of the methods are primal feasible at initialization, MARA achieve better suboptimality quickly as well. In particular, MARA with $\epsilon_v = 10\%$ and $\epsilon_p = 10\%$ produces primal feasible points at iterations 22 and 20 with function value suboptimality of 3.7% and 3.2%, respectively. In contrast, the competing methods have suboptimality ranging from 23% to 130%.

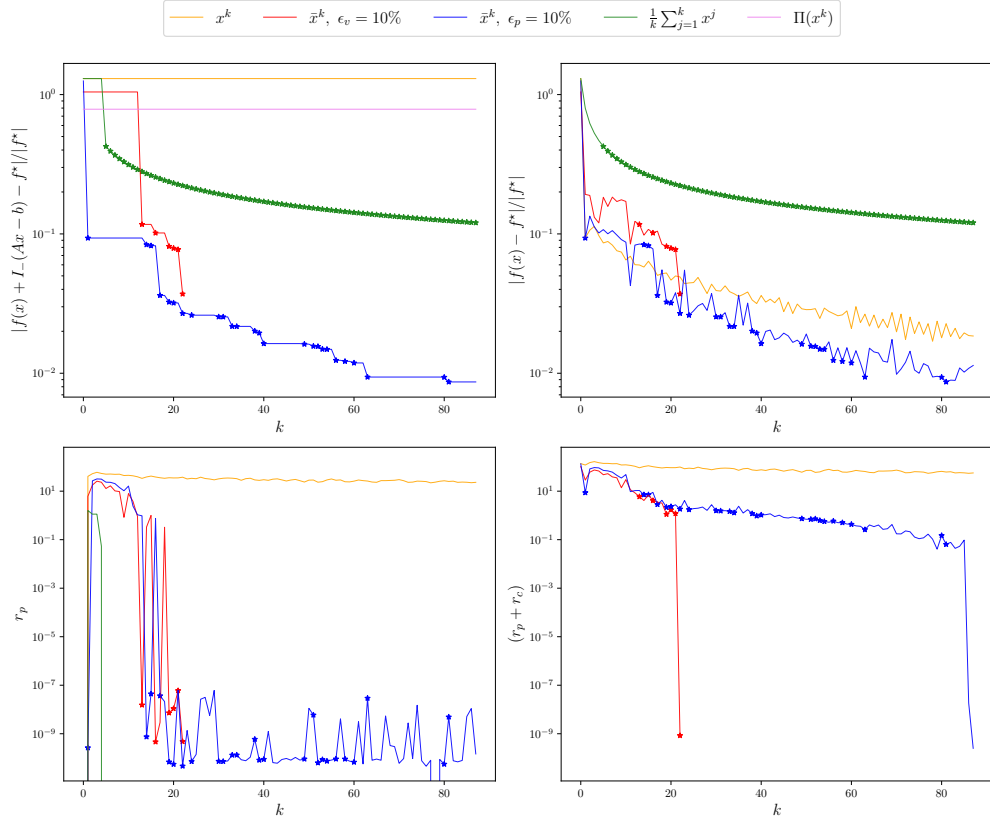


Figure 5: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), primal violations (bottom left), residuals (bottom right) versus dual subgradient method iterations for convex relaxation of the assignment problem.

6.5 Multi-commodity flow problem

We consider a variation of the multi-commodity flow problem from [PZB24, §4.3]. Let $A \in \mathbf{R}^{p \times q}$ be the incidence matrix of a graph with p vertices or nodes and q directed edges. The network supports the flow of K different commodities. Each commodity $i = 1, \dots, K$ is shipped from a source node $r_i \in \{1, \dots, p\}$ to a sink $s_i \in \{1, \dots, p\}$ over the network edges with total flow given by $d_i \geq 0$. We also define $z_i \in \mathbf{R}_+^q$ as the vector of flows of commodity i on the edges. Then the flow conservation is given by

$$Az_i + d_i(e_{r_i} - e_{s_i}) = 0, \quad i = 1, \dots, K,$$

where e_j is the j th unit vector.

We consider the multi-commodity problem

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^K b_i \sqrt{d_i} \\ & \text{subject to} && 0 \leq z_i \leq x_i, \quad i = 1, \dots, K \\ & && Az_i + d_i(e_{r_i} - e_{s_i}) = 0, \quad i = 1, \dots, K \\ & && v_1 x_1 + \dots + v_K x_K \leq c, \quad x_i \geq 0, \quad i = 1, \dots, K, \end{aligned}$$

where z_i , d_i , and x_i are variables, $c \in \mathbf{R}_+^q$ is a given vector of edge capacities, and $v \in \mathbf{R}_+^K$ is a given vector of commodity volumes.

We define the agent function $f_i(x_i)$ as the optimal value of the single commodity problem

$$\begin{aligned} & \text{minimize} && -b_i \sqrt{d_i} \\ & \text{subject to} && 0 \leq z_i \leq x_i \\ & && Az_i + d_i(e_{r_i} - e_{s_i}) = 0 \\ & && x_i \leq R\mathbf{1}, \end{aligned}$$

with variables z_i and d_i , for all $i = 1, \dots, K$. Then the distributed optimization problem takes the form

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^K f_i(x_i) \\ & \text{subject to} && v_1 x_1 + \dots + v_K x_K \leq c. \end{aligned}$$

Problem instance We consider an example with $K = 100$ commodities, and a graph with $p = 15$ nodes and $q = 100$ edges. Edges are generated randomly from pairs of nodes, with an additional cycle passing through all vertices (to ensure that the graph is strongly connected, *i.e.*, there is a directed path from any node to any other). We randomly choose the source-destination pairs (r_i, s_i) . We choose commodity capacities $\tilde{c}_j$ from a uniform distribution on $[0.2, 2]$, and b_i are chosen uniformly on $[0.5, 1.5]$. Volumes are sampled from $\exp N(0, 1)$. Then the capacities are set as expected volume per edge, *i.e.*, $c_j = \tilde{c}_j \exp(1/2)$ for all $j = 1, \dots, q$. The dimension of primal variable is 10000 and of dual variable is 100.

Results Figure 6 illustrates suboptimality and relative residuals achieved with the localization method for the multi-commodity flow problem. At iteration 62, MARA with $\epsilon_v = 10\%$

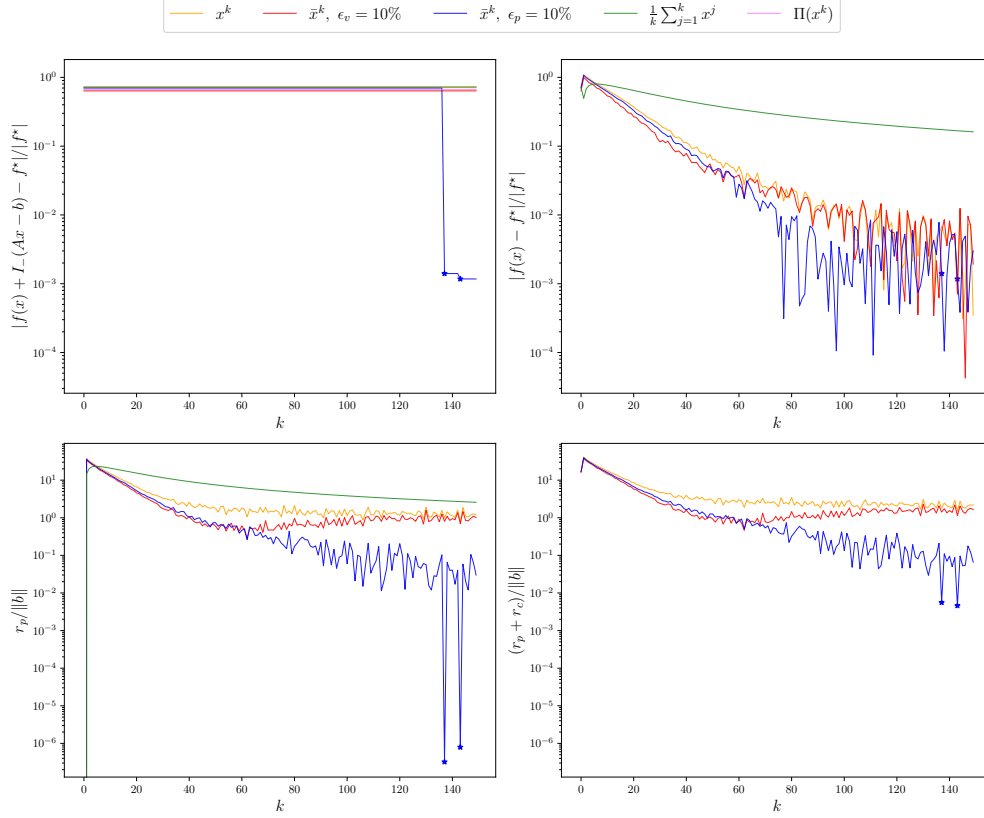


Figure 6: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the multi-commodity problem.

finds a primal point $\bar{x}^k$ with 28% relative primal infeasibility, 1.8% function value suboptimality, and $2.8\times$ smaller primal residual than x^k . At iteration 137 MARA with $\epsilon_p = 10\%$ finds a primal feasible point with 0.1% function value suboptimality.

In contrast, at iteration 143, x^k is primal infeasible with 79% relative infeasibility and 0.1% suboptimality. At iteration 149, the primal average value is infeasible with 258% relative infeasibility and 16% suboptimality. Moreover, except for the first iteration, the projected point $\Pi(x^k)$ is everywhere outside the domain of the agents' functions. The competing methods are primal feasible only at the initial iteration with the following suboptimality levels: the primal point x^k is 72% suboptimal, primal average point $\frac{1}{k} \sum_{j=1}^k x^j$ is 72% suboptimal, and the projected points $\Pi(x^k)$ is 67% suboptimal.

Similarly, Figure 7 illustrates suboptimality and relative residuals achieved with the dual subgradient method. MARA with both approximate oracles recovers primal feasible points with suboptimality levels that are a factor $\times 5$ to $\times 7$ lower than those of the competing methods.

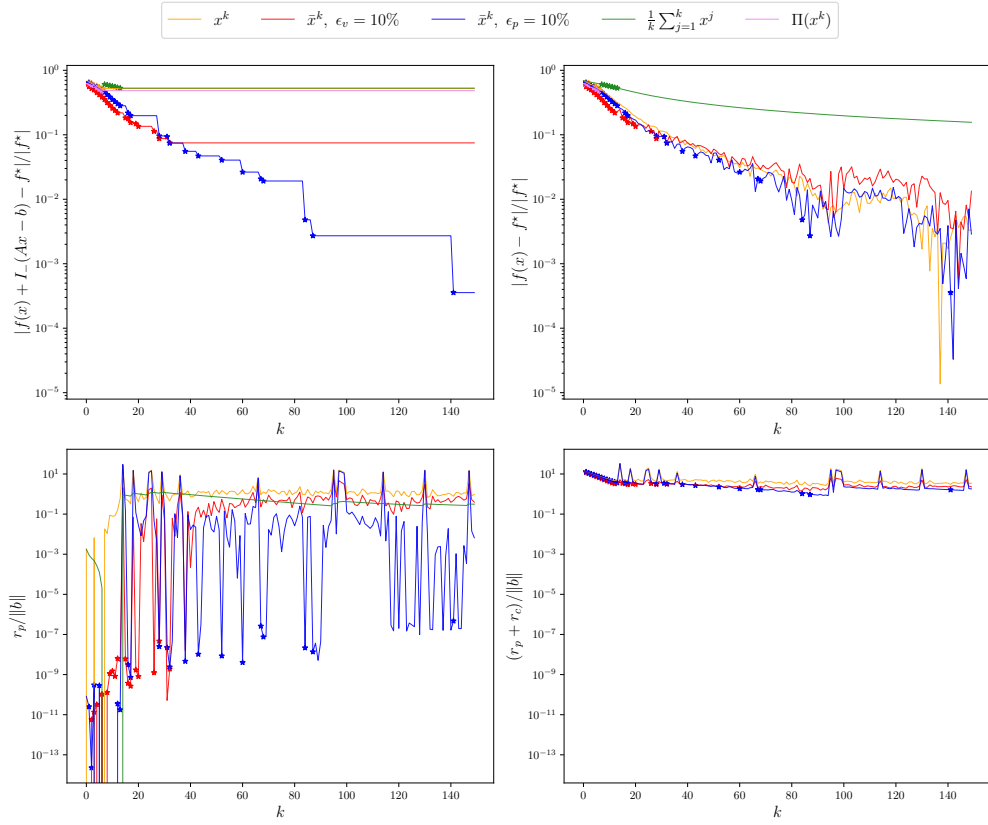


Figure 7: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus dual subgradient method iterations for the multi-commodity problem.

6.6 Shipment problem

We consider a shipment problem given by the following capacitated optimal transport formulation

$$\begin{aligned} & \text{minimize} && \text{Tr}(C^T X) \\ & \text{subject to} && X\mathbf{1} = \mu_s \\ & && X_{i,j} \geq 0, \quad i = 1, \dots, K, \quad j = 1, \dots, m \\ & && X^T \mathbf{1} = \mu_t \\ & && X^T v \leq c \end{aligned}$$

where $C \in \mathbf{R}_+^{K \times m}$ is a cost matrix, $\mu_s \in \mathbf{R}^K$ is a source distribution, $\mu_t \in \mathbf{R}^m$ is a target distribution, $v \in \mathbf{R}^K$ is a vector with volumes, $c \in \mathbf{R}^m$ is a vector with capacities, and $X \in \mathbf{R}_+^{K \times m}$ is a variable.

Let $C_i \in \mathbf{R}^m$ and $x_i \in \mathbf{R}^m$ denote the i -th row of C and X , respectively. We define the agent function $f_i : \mathbf{R}^m \rightarrow \mathbf{R} \cup \{\infty\}$ as

$$f_i(x_i) = \begin{cases} C_i^T x_i & x_i \geq 0, \quad x_i^T \mathbf{1} = (\mu_s)_i \\ \infty & \text{otherwise} \end{cases}$$

for all $i = 1, \dots, K$. Then the distributed optimization problem takes the form

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^K f_i(x_i) \\ & \text{subject to} && X^T \mathbf{1} = \mu_t \\ & && X^T v \leq c \\ & && X^T = [x_1 \quad \dots \quad x_K]. \end{aligned}$$

Problem instance We sample entries of source and target locations, $s \in \mathbf{R}^{K \times d}$ and $t \in \mathbf{R}^{m \times d}$, from the standard normal. Then entries of cost matrix are defined as $C_{ij} = \|s_i - t_j\|_2^2$ for all $i = 1, \dots, K$ and $j = 1, \dots, m$, where s_i and t_j are normalized to unit vectors. The source and target distributions μ_s and μ_t respectively are sampled from log-normal and normalized to sum to one. Volumes are sampled from $\exp N(0, \sigma^2)$. Then capacities are set as expected volume per target location, *i.e.*, $c_j = (\mu_t)_j \exp(\sigma^2/2)$ for all $j = 1, \dots, m$.

For this specific instance, we consider $m = 25$, $K = 100$ agents, $d = 10$ and $\sigma = 0.8$. The dimension of primal variable is 2500 and the dual variable has dimension 50.

Results Figure 8 illustrates suboptimality and relative residuals achieved with localization method for shipment problem. At iteration 39, MARA with $\epsilon_v = 10\%$ finds a feasible primal point $\bar{x}^k$ with 6.3% function value suboptimality. At iteration 93 MARA with $\epsilon_p = 10\%$ finds a primal point $\bar{x}^k$ with 0.3% function value suboptimality and $8\times$ reduction of primal violation relative to x^k and $4\times$ reduction relative to the primal average point.

In contrast, at iteration 77, x^k is primal infeasible with 86% relative infeasibility and 6.6% suboptimality. At iteration 99, the primal average point is infeasible with 42% relative infeasibility and 7% suboptimality. The projected point $\Pi(x^k)$ is everywhere outside the domain of the agents' objective functions.

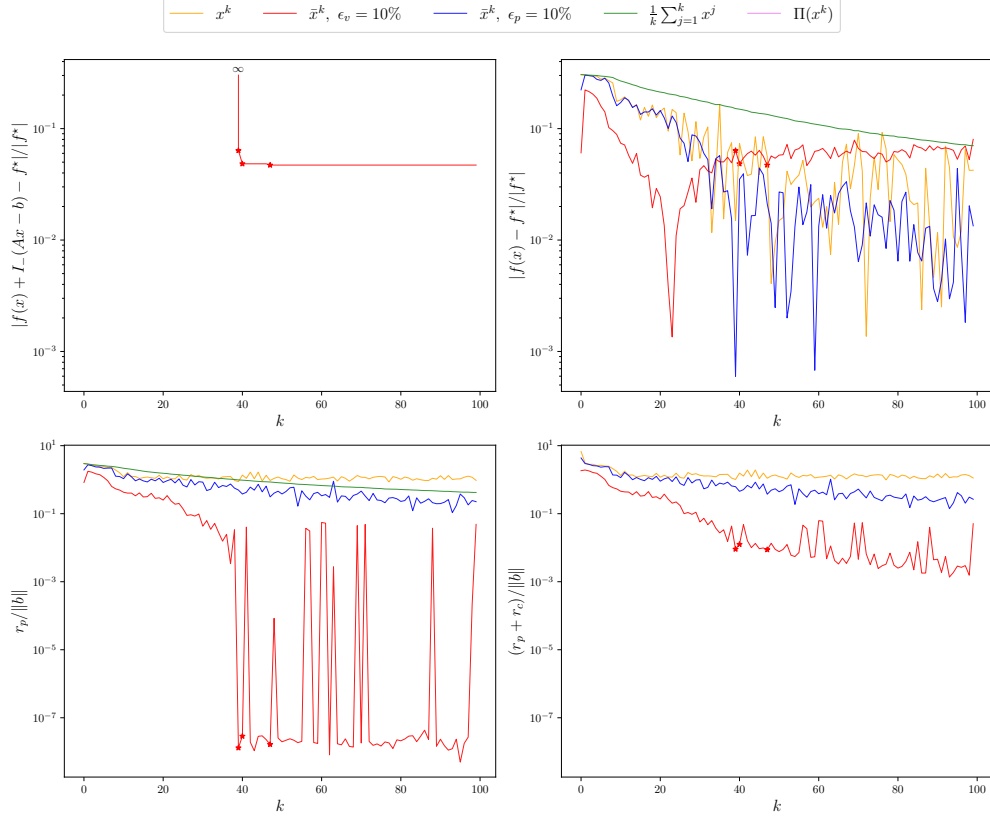


Figure 8: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the shipment problem.

Similarly, Figure 9 illustrates suboptimality and relative residuals achieved with the dual subgradient method. MARA with $\epsilon_v = 10\%$ recovers a primal feasible point at iteration 16 with suboptimality of 15.5%, while the MARA price method with $\epsilon_p = 10\%$ fails to find a feasible solution. All competing methods are unable to recover a primal feasible point, with relative infeasibility ranging from 26% to 87% and suboptimality ranging from 1.1% to 6.5%.

6.7 Approximation trade-offs

Next we evaluate the performance of MARA using the price localization method by varying two factors: 1) oracle suboptimality, $\epsilon = 1\%$ and $\epsilon = 10\%$, and 2) number of responses, $N = 10$ and $N = 50$.

Oracle suboptimality ϵ Recall from §4.1, that the oracle’s suboptimality has additive error term proportional to ϵ . Thus, we expect smaller values of ϵ to improve the function value suboptimality of MARA. At the same time primal points generated by the approximate

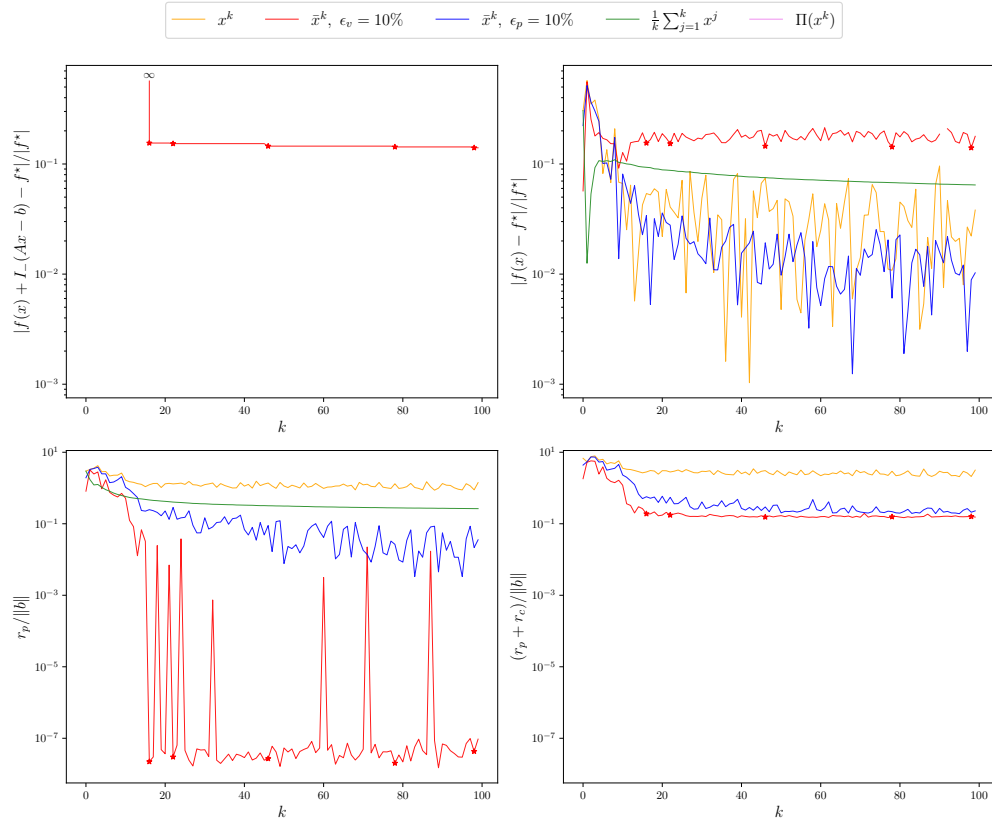


Figure 9: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus dual subgradient method iterations for the shipment problem.

oracle with smaller ϵ are a subset of those with larger ϵ , which implies slower reduction in residuals.

This behavior is indeed observed in our numerical experiments; see Figures 10, 11, 13, and 12. For example, as shown in Figure 10, MARA with $\epsilon_p = 10\%$ finds a primal feasible primal point at iteration 94, much faster than the method with $\epsilon_p = 1\%$, which takes 165 iterations. Their best suboptimality for primal feasible points are 0.1% and 0.005% respectively. Similarly, MARA with $\epsilon_v = 10\%$ finds a feasible primal point at iteration 0, while the method $\epsilon_v = 1\%$ takes 45 iterations. However, the best suboptimality of the feasible points for the two methods are comparable, 0.16% and 0.22%, respectively.

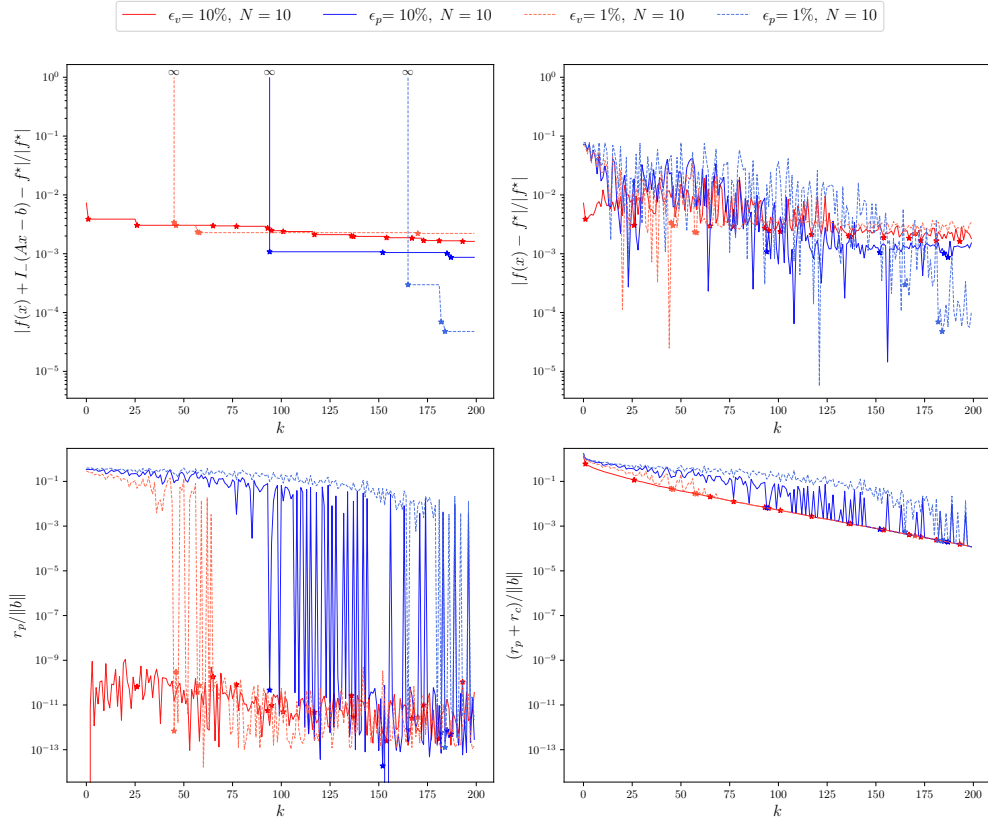


Figure 10: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the resource allocation problem with $\epsilon = \{1\%, 10\%\}$ and $N = 10$.

Number of responses N As discussed in §4.1, the primal point returned by the primal recovery method is a convex combination of N_i ϵ -suboptimal primal points for each agent $i = 1, \dots, K$. Increasing N_i therefore enlarges the convex hull of the candidate primal points, implying a faster reduction in residuals.

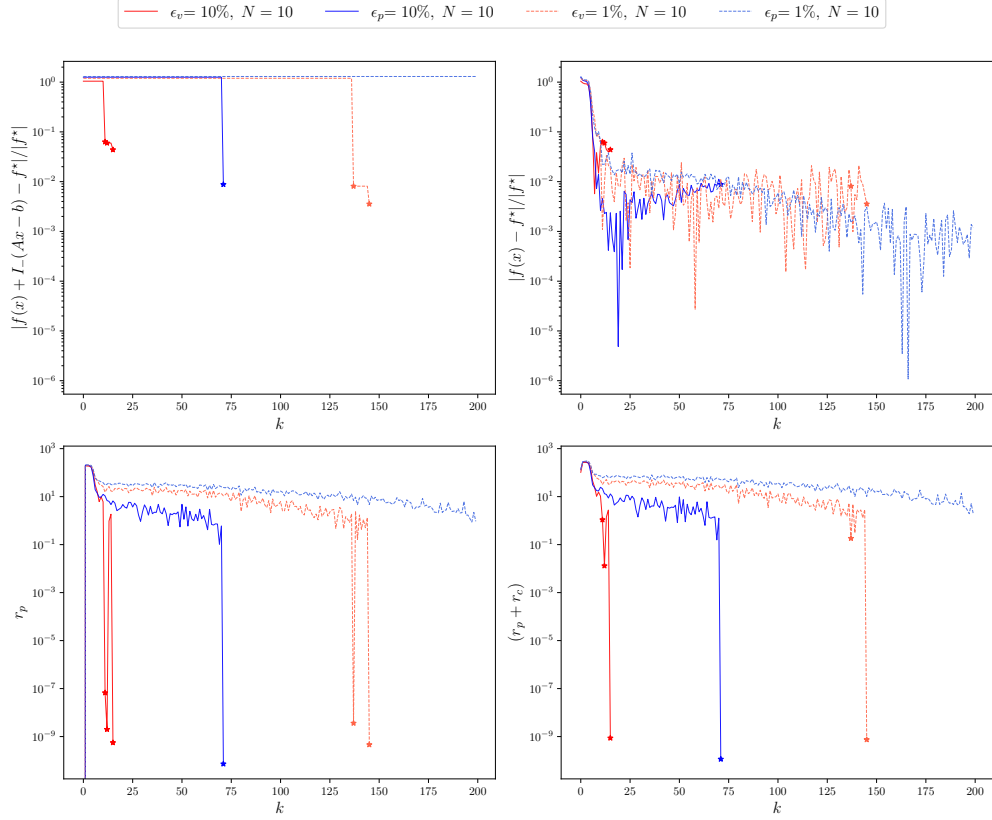


Figure 11: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), primal violations (bottom left), residuals (bottom right) versus localization method iterations for convex relaxation of the assignment problem for $\epsilon = \{1\%, 10\%\}$ and $N = 10$.

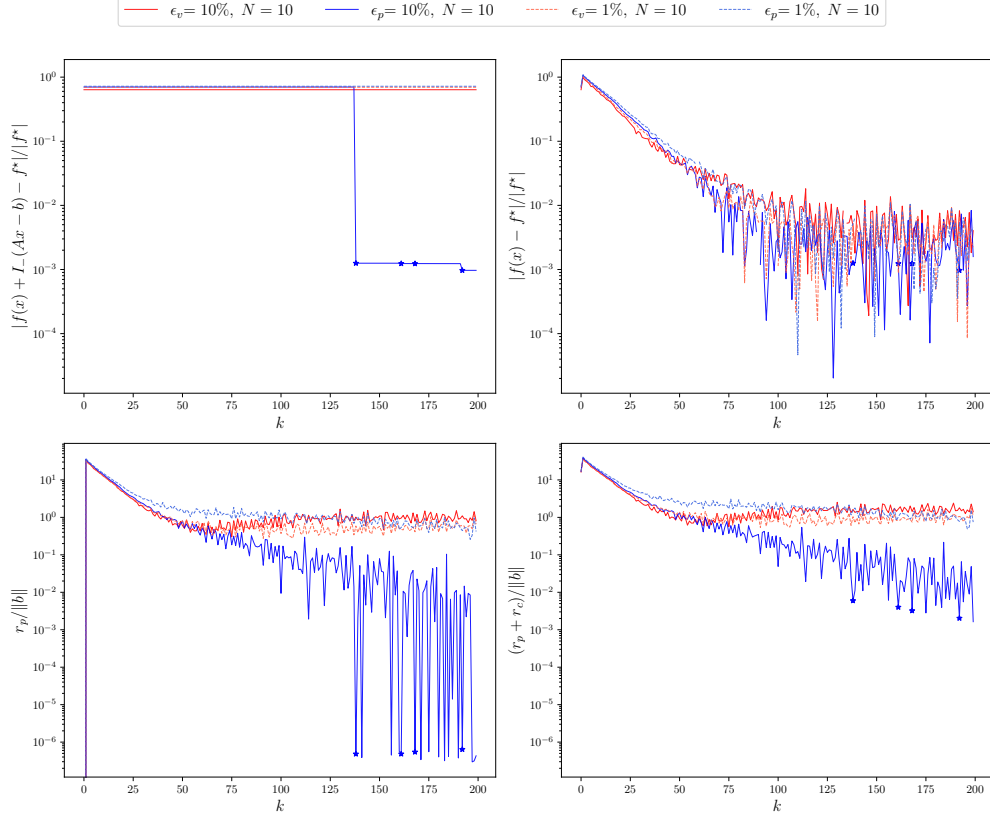


Figure 12: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the multi-commodity flow problem with $\epsilon = \{1\%, 10\%\}$ and $N = 10$.

This behavior is observed in our numerical experiments, see Figures 14, 15, 17 and 16. For example, as shown in Figure 16, (excluding the initial iteration), MARA with $\epsilon_p = 10\%$ and $N = 10$ finds a primal feasible primal point at iteration 137, much slower than the method with $\epsilon_p = 10\%$ and $N = 50$, which takes 81 iterations. Their best suboptimality for feasible points for two methods are comparable, 0.10% and 0.13%, respectively. The results of this section suggest it might be beneficial to split the budget of samples N_i across oracles with different suboptimality.

6.8 Including history in MARA

In this section, we test using historical data for MARA primal recovery while solving problem (13), as explained in §5. We used the localization method for price discovery. Our experiments demonstrate that incorporating history H accelerates the identification of primal feasible points, with a slight increase in suboptimality. See Figures 18, 19, 20, and 21. For

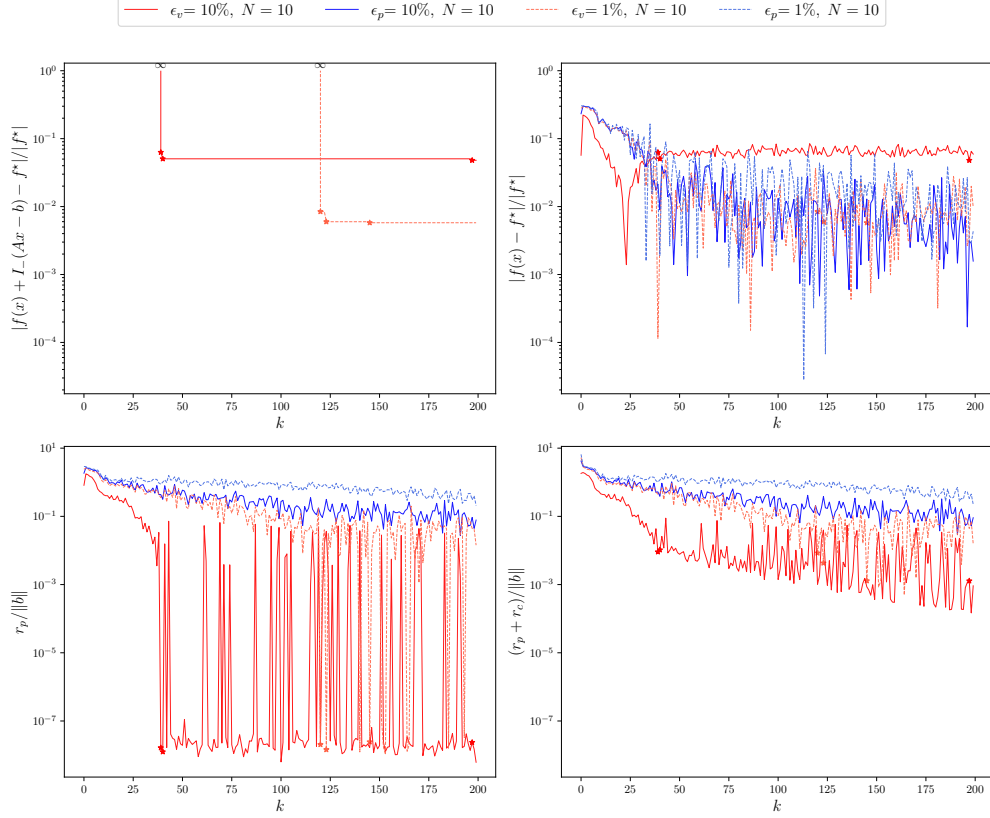


Figure 13: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the shipment problem with $\epsilon = \{1\%, 10\%\}$ and $N = 10$.

example, in the resource allocation problem $\epsilon_p = 10\%$ primal recovery method with history $H = 3$ finds a feasible point at iteration 37 with suboptimality 0.25%, instead of iteration 92 with suboptimality 0.15% without history ($H = 1$), see Figures 18 and 2, respectively.

Similarly, in the multi-commodity flow problem without history, the $\epsilon_v = 10\%$ MARA does not recover the primal feasible point except the initial iteration, while $\epsilon_p = 10\%$ MARA recovers primal feasible point at iteration 137 with suboptimality 0.14%. In contrast, with history $H = 5$, MARA with both dual methods recovers primal feasible points faster, at iterations 68 and 78, with suboptimality 2.7% and 1.1%, respectively. See Figures 6 and 20.

7 Conclusions

MARA works with any dual optimization algorithm, providing a flexible way to construct fast, feasible, and controllably-suboptimal solutions. Numerical examples on several problem

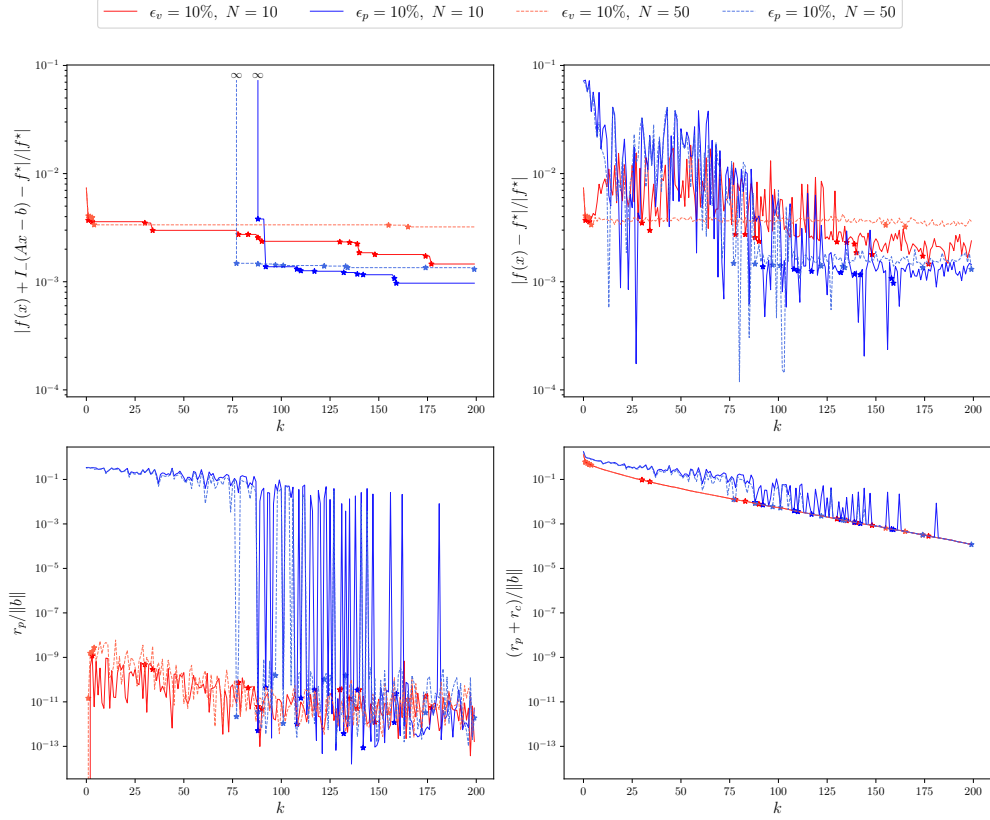


Figure 14: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the resource allocation problem with $N = \{10, 50\}$ and $\epsilon = 10\%$.

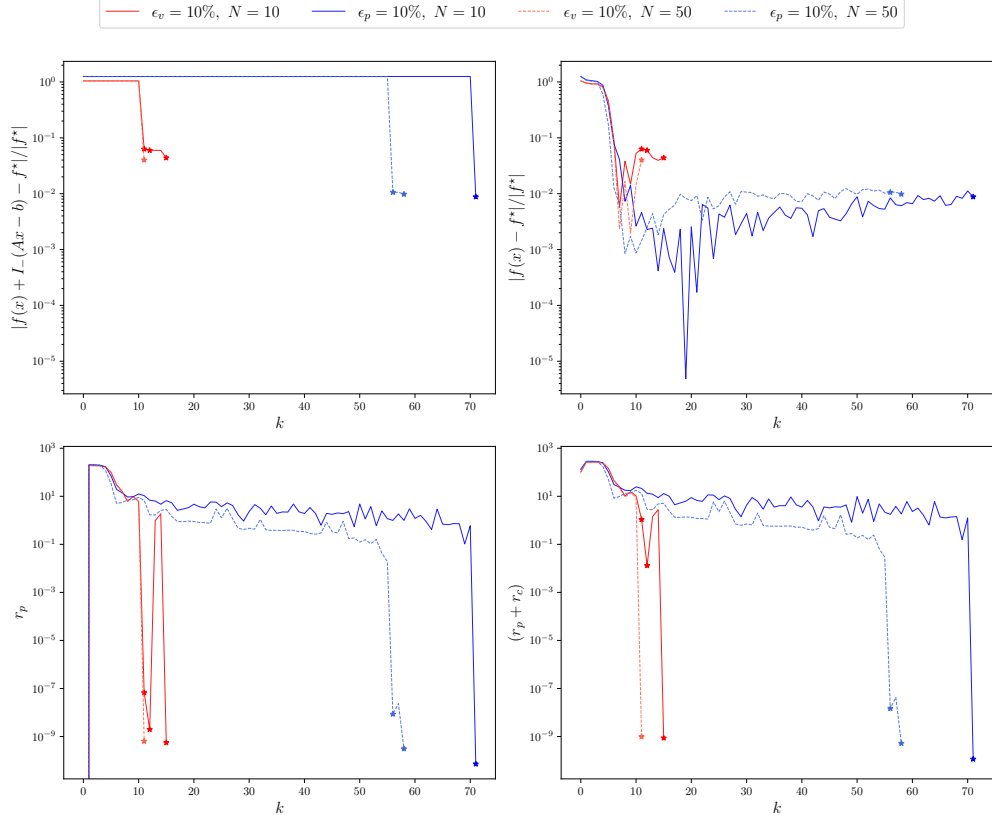


Figure 15: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for convex relaxation of the assignment problem with $N = \{10, 50\}$ and $\epsilon = 10\%$.

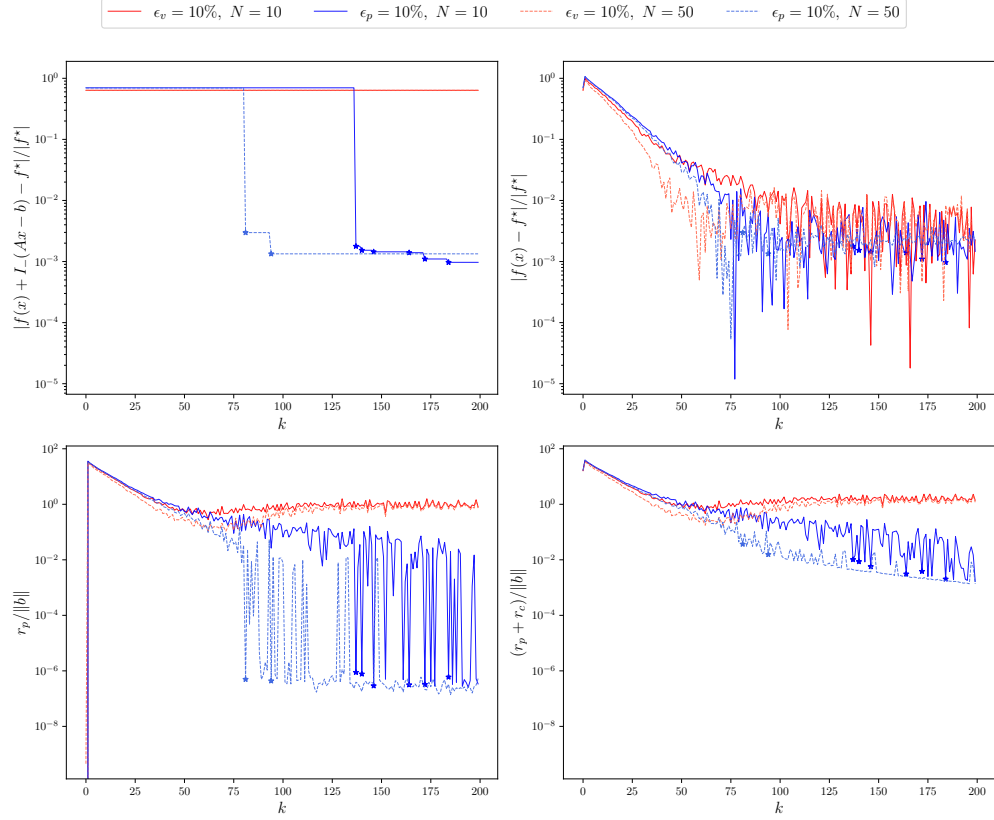


Figure 16: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the multi-commodity flow problem with $N = \{10, 50\}$ and $\epsilon = 10\%$.

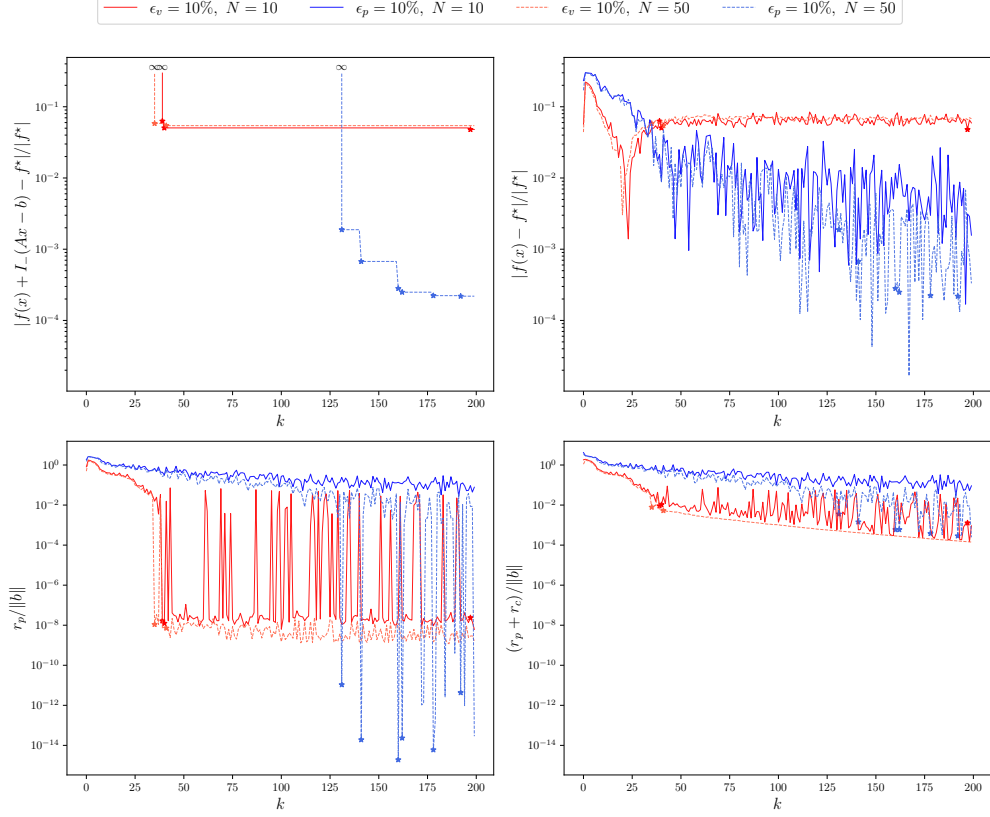


Figure 17: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the shipment problem for $N = \{10, 50\}$ and $\epsilon = 10\%$.

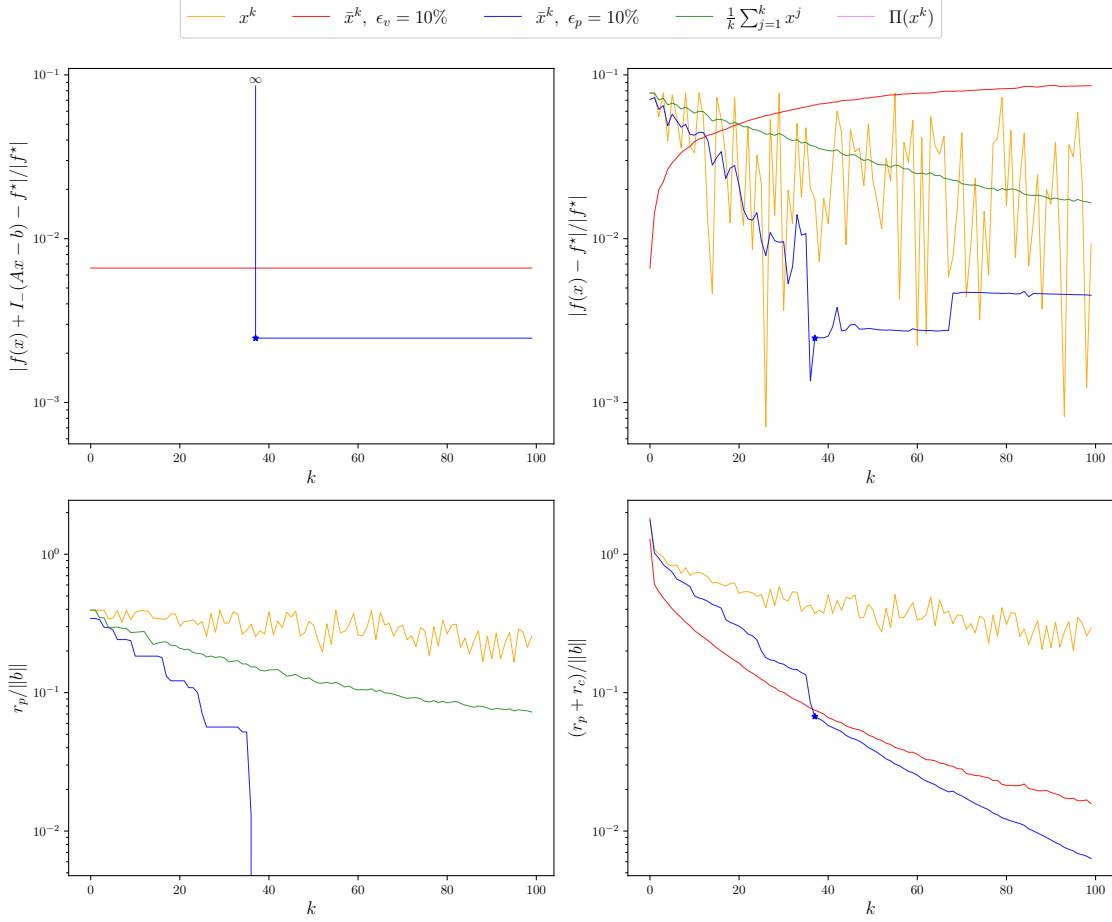


Figure 18: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the resource allocation problem with history $H = 3$.

classes using both a price localization method and a dual subgradient method show that MARA is able to generate feasible, near-optimal solutions in just a few tens of iterations of the underlying dual algorithm. While MARA increases total computation budgets significantly, because it operates independently of the dual algorithm using parallel calls to agents, it does not impact the wall-clock time of the underlying dual algorithm. With the increasing scale of computing resource available in modern data centers, MARA’s capability to leverage a large computing budget to quickly find feasible, near-optimal solutions for large-scale problems – while providing hyperparameters to control the speed and solution quality trade-off – is valuable.

MARA is flexible, so further research will no doubt uncover other variations that can further improve its performance. Another useful extension is to non-convex agents, in particular agents with integrality constraints for which primal recovery is especially challenging.

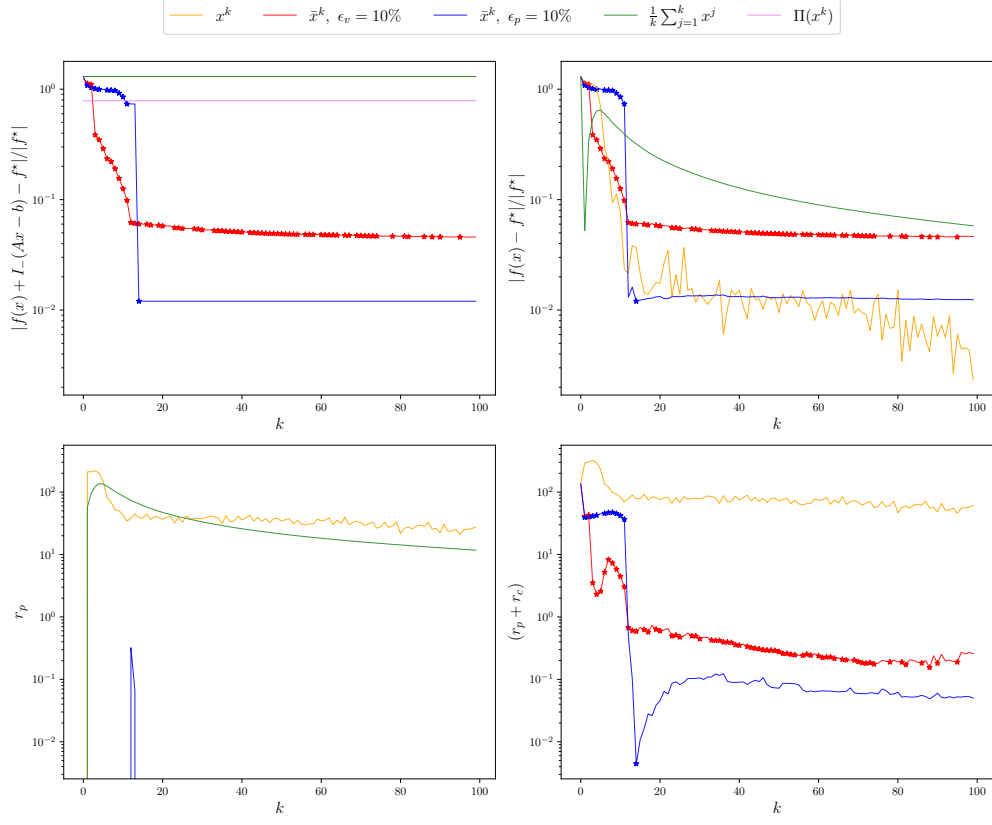


Figure 19: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for convex relaxation of the assignment problem with history $H = 3$.

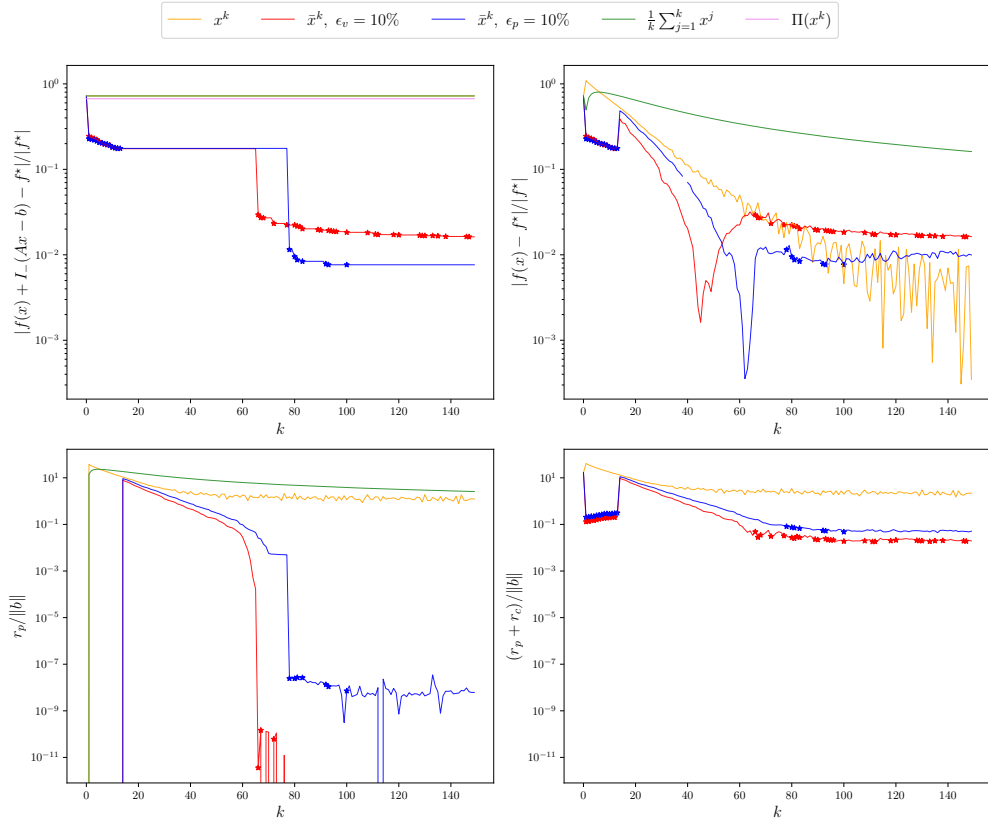


Figure 20: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the multi-commodity problem with history $H = 5$.

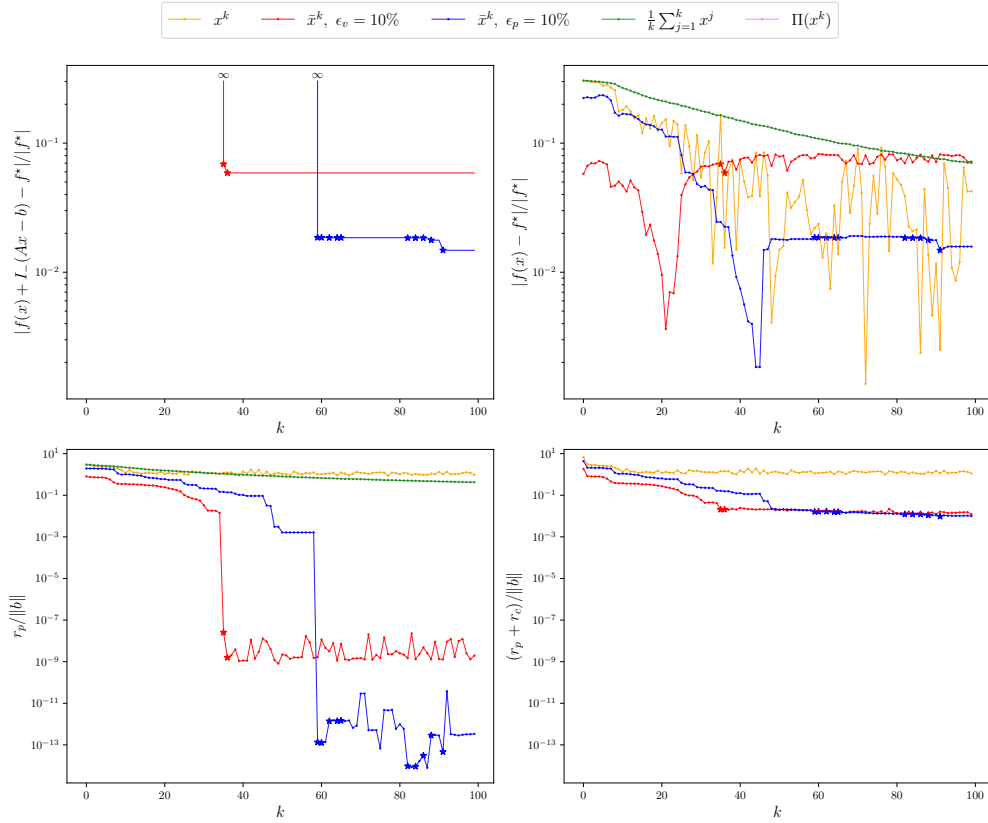


Figure 21: Best-to-date suboptimality of the most recent feasible point (top left), function value suboptimality (top right), relative primal violations (bottom left), relative residuals (bottom right) versus localization method iterations for the shipment problem with history $H = 3$.

In this case, the MARA step of taking convex combinations of responses can be replaced by an integer program to select one response from each agent. (See Appendix [B](#).) This is an important practical extension on which we are actively working.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Acknowledgements

Not applicable.

References

- [AW13] T. Achterberg and R. Wunderling. Mixed integer programming: Analyzing 12 years of progress. In *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pages 449–481. Springer, 2013.
- [Ber99] D. Bertsekas. *Nonlinear Programming*. 1999.
- [Ber09] D. Bertsekas. *Convex Optimization Theory*, volume 1. Athena Scientific, 2009.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [BV07] S. Boyd and L. Vandenberghe. Localization and cutting-plane methods. *Stanford EE 364b lecture notes*, 386, 2007.
- [BVS08] S. Boyd, L. Vandenberghe, and J. Skaf. Analytic center cutting-plane method. *Lecture Notes from Stanford University*, 2008.
- [BXM03] S. Boyd, L. Xiao, and A. Mutapcic. Subgradient methods. *Lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004(01), 2003.
- [CFM75] P. Camerini, L. Fratta, and F. Maffioli. On improving relaxation methods by modified gradient techniques. *Nondifferentiable Optimization*, pages 26–34, 1975.
- [CJOS24] Y. Carmon, A. Jambulapati, L. O’Carroll, and A. Sidford. Extracting dual solutions via primal optimizers. *arXiv preprint arXiv:2412.02949*, 2024.
- [EM75] J. Elzinga and T. Moore. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.
- [GV93] J.-L. Goffin and J.-P. Vial. On the computation of weighted analytic centers and dual ellipsoids with the projective algorithm. *Mathematical Programming*, 60(1):81–92, 1993.
- [LL97] T. Larsson and Z. Liu. A lagrangean relaxation scheme for structured linear programs with application to multicommodity network flows. *Optimization*, 40(3):247–284, 1997.
- [LL99] S. Low and D. Lapsley. Optimization flow control. i. basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.
- [LPS99] T. Larsson, M. Patriksson, and A.-B. Strömberg. Ergodic, primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming*, 86:283–312, 1999.
- [Nes13] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

- [NJ78] A. Nemirovskii and D. Judin. Cezare convergence of gradient method approximation of saddle points for convex-concave functions. *Doklady Akademii Nauk SSSR*, 239:1056–1059, 1978.
- [NO09] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009.
- [NV99] Y. Nesterov and J.-P. Vial. Homogeneous analytic center cutting plane methods for convex problems and variational inequalities. *SIAM Journal on Optimization*, 9(3):707–728, 1999.
- [Pét02] O. Péton. *The homogeneous analytic center cutting plane method*. PhD thesis, Verlag Nicht Ermittellbar, 2002.
- [Pol87] B. Polyak. *Introduction to Optimization*. Optimization Software, New York, 1987.
- [PZB24] T. Parshakova, F. Zhang, and S. Boyd. Implementation of an oracle-structured bundle method for distributed optimization. *Optimization and Engineering*, 25(3):1685–1718, 2024.
- [RN23] A. Rodomanov and Y. Nesterov. Subgradient ellipsoid method for nonsmooth convex problems. *Mathematical Programming*, 199(1):305–341, 2023.
- [SB10] J. Skaf and S. Boyd. Techniques for exploring the suboptimal set. *Optimization and Engineering*, 11(2):319–337, 2010.
- [Sho85] N. Shor. *Minimization Methods for Nondifferentiable Functions*. Springer, Berlin, 1985. Translated from Russian by K. C. Kiwiel and A. Ruszczyński.
- [Sho87] N. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.
- [Sho12] N. Shor. *Minimization Methods for Non-differentiable Functions*, volume 3. Springer Science & Business Media, 2012.
- [SJR16] A. Simonetto and H. Jamali-Rad. Primal recovery from consensus-based dual decomposition for distributed convex optimization. *Journal of Optimization Theory and Applications*, 168:172–197, 2016.
- [Son88] G. Sonnevend. New algorithms in convex programming based on a notion of “centre” (for systems of analytic inequalities) and on rational extrapolation. In *Trends in Mathematical Optimization: 4th French-German Conference on Optimization*, pages 311–326. Springer, 1988.
- [SS86] S. Sen and H. Serali. A class of convergent primal-dual subgradient algorithms for decomposable convex programs. *Mathematical Programming*, 35:279–297, 1986.

- [ST20] H. Santos and T. Toffolo. Mixed integer linear programming with Python. *COINOR Computational Infrastructure for Operations Research*, 2020.
- [Tar88] S. Tarasov. The method of inscribed ellipsoids. In *Soviet Mathematics-Doklady*, volume 37, pages 226–230, 1988.
- [Vai89] P. Vaidya. A new algorithm for minimizing convex functions over convex sets. In *30th Annual Symposium on Foundations of Computer Science*, pages 338–343. IEEE Computer Society, 1989.
- [Wan17] Mengdi Wang. Vanishing price of decentralization in large coordinative nonconvex optimization. *SIAM Journal on Optimization*, 27(3):1977–2009, 2017.

A Homogeneous ACCPM

The following properties are based on paper [NV99]. For any $z = (\bar{\lambda}, t) \in \mathbf{R}_+^m \times \mathbf{R}_{++}$, define $\lambda = \bar{\lambda}/t$ and

$$\hat{q}(z) = (-g'(\lambda), g'(\lambda)^T \lambda) \in \mathbf{R}^{m+1}, \quad (-g)'(\lambda) = b - Ax(-A^T \lambda) \in \partial(-g)(\lambda).$$

Then $q(z) = \hat{q}(z)/\|\hat{q}(z)\|_2$ is a homogeneous separation oracle with norm one.

Let the normal log-barrier be defined as

$$F^{k+1}(z) = F^k(z) - \log(q(z_k)^T(z_k - z)).$$

Define the following quantities

$$\beta_{ik} = \frac{1}{q(z_i)^T(z_i - z_k)} > 0, \quad i = 1, \dots, k, \quad S_k = \sum_{i=1}^k \beta_{ik},$$

and the weighted average of slacks at z as

$$\mu_k(z) = \frac{1}{S_k} \sum_{i=1}^k \beta_{ik} q(z_i)^T(z_i - z).$$

Then by Theorem 2.7 in [NV99] we have

$$\mu_k(z) \leq O(1/\sqrt{k}).$$

Also define the quantities

$$\pi_{ik} = \beta_{ik}/\|\hat{q}(z_i)\|_2, \quad i = 1, \dots, k, \quad P_k = \sum_{i=1}^k \pi_{ik}.$$

Using Lemma 4.3 in [NV99], we have

$$\hat{q}(z)^T(z - \tilde{z}) = \tilde{t}(-g)'(\lambda)^T(\lambda - \tilde{\lambda}).$$

Then for any $\lambda \in \mathbf{R}_+^m$, and $z = (\lambda, 1)$, it follows

$$\frac{1}{P_k} \sum_{i=1}^k \pi_{ik} (-g)'(\lambda_i)^T(\lambda_i - \lambda) = \frac{1}{P_k} \sum_{i=1}^k \pi_{ik} \hat{q}(z_i)^T(z_i - z) = \frac{S_k}{P_k} \mu_k(z) = O(1/\sqrt{k}),$$

see [NV99]. Define $\theta_{ik} = \pi_{ik}/P_k$ (clearly, $\sum_{i=1}^k \theta_{ik} = 1$), this implies

$$\begin{aligned} O(1/\sqrt{k}) &\geq \sum_{i=1}^k \theta_{ik} (-g)'(\lambda_i)^T(\lambda_i - \lambda^*) \\ &\geq g(\lambda^*) - \sum_{i=1}^k \theta_{ik} g(\lambda_i) \\ &\geq g(\lambda^*) - g(\bar{\lambda}_k), \end{aligned}$$

where $\bar{\lambda}_k = \sum_{i=1}^k \theta_{ik} \lambda_i$. In the above we used concavity of the dual function g and Jensen inequality. Combining the above with $g(\lambda^*) \geq g(\bar{\lambda}_k)$, we get $g(\bar{\lambda}_k) \rightarrow g(\lambda^*)$.

B Nonconvex agents

MARA can be extended to the settings where the problem (1) is nonconvex. We assume that each agent has access to the price-directed oracle (4). Since the dual problem is always convex, any price discovery method conforming to our access assumptions (*e.g.*, see §6.1.2) can be used directly to get optimal dual prices. However, since the function f is nonconvex, strong duality rarely holds.

B.1 Lagrangian relaxation

Since weak duality always holds, the dual problem is a Lagrangian relaxation of the non-convex primal problem. Lagrangian relaxation has been studied since 1980s [Sho87], for more results on duality see [BV04, §5] and [Ber99, §5]. The dual problem associated with problem (1) is given by

$$\begin{aligned} & \text{maximize} && -f^*(u) - \lambda^T b \\ & \text{subject to} && u = -A^T \lambda \\ & && \lambda \geq 0. \end{aligned} \tag{16}$$

For $\lambda \geq 0$ and dual variable $\hat{x}$ associated with the equality constraints, the Lagrangian of (16) is

$$L(\lambda, u, \hat{x}) = -f^*(u) - \lambda^T b + \hat{x}^T (u + A^T \lambda).$$

Thus the Lagrange dual problem associated with (16) is

$$\begin{aligned} & \text{minimize} && f^{**}(\hat{x}) \\ & \text{subject to} && A\hat{x} \leq b. \end{aligned} \tag{17}$$

By the conjugacy theorem [Ber09], the double conjugate function $f^{**}(\hat{x})$ is a convex closure of function f , and since f^* is a closed, convex, and proper function, we have

$$(f^{**})^* = (f^*)^{**} = f^*.$$

Therefore, assuming that there exists $\hat{x} \in \mathbf{relint\,dom}\,f^{**}$ with $A\hat{x} \leq b$, then by Slater's condition strong duality holds between problems (17) and (16).

B.2 Nonconvex primal recovery

Due to the strong duality between problems (17) and (16), the optimality conditions for problem (17) are also (5)–(8). Problem (17) is a convex relaxation of the nonconvex problem (1), thus we generate a primal variable reducing the primal residuals r_p as follows. Similar to §4.3, each agent $i = 1, \dots, K$ provides a list of N_i ϵ -suboptimal primal variables, $\{z_{i1}, \dots, z_{iN_i}\}$, associated with the current local price vector $y_i = -A_i^T \lambda$. Since the functions f_i are nonconvex, to produce primal variable $\bar{x}_i$ in the domain of f_i , we select a point from

the set $\{z_{i_1}, \dots, z_{N_i}\}$, for all $i = 1, \dots, K$. In particular, we solve the following mixed-integer linear program (MILP)

$$\begin{aligned}
& \text{minimize} && r_p \\
& \text{subject to} && r_p = \mathbf{1}^T (A\bar{x} - b)_+ \\
& && \bar{x}_i = Z_i u_i, \quad i = 1, \dots, K \\
& && \mathbf{1}^T u_i = 1, \quad i = 1, \dots, K \\
& && u_i \in \{0, 1\}^{N_i}, \quad i = 1, \dots, K \\
& && \bar{x} = (\bar{x}_1, \dots, \bar{x}_K),
\end{aligned} \tag{18}$$

where u_i is a variable, and Z_i is defined as in (12), for all $i = 1, \dots, K$. The problem (18) can be readily solved using any MILP solver [AW13, ST20]. Since this MILP may be computationally expensive, we next describe a simple heuristic for solving (18).

B.3 MILP heuristic

Relax (18) by replacing each nonconvex constraint $u_i \in \{0, 1\}^{N_i}$ with its convex relaxation $u_i \in [0, 1]^{N_i}$ for all $i = 1, \dots, K$. In the relaxed solution, interpret each component u_{ij} as the success probability of the corresponding Bernoulli variable for all $j = 1, \dots, N_i$. Next, we generate S samples from these Bernoulli distributions and apply a greedy rounding method to each sample. Finally, we select the primal variable with the lowest residual as the approximate solution. We summarize the greedy rounding method:

1. Begin with an initial set of fractional variables.
2. Successively iterate through each variable.
 - (a) For each variable, test the possible values $\{0, 1\}$ while keeping the other variables fixed.
 - (b) If any value results in a lower objective function, select that value and proceed to the next variable.
3. Continue cycling until a local minimum is reached.

Note that step 2(a) can be carried out efficiently for the objective (18).