International Journal of Electronics and Telecommunication Systems
Volume1, Issue 02 2025

Synchronous and asynchronous Bertsekas methods for energy-aware networks and solution of Regularized Linear Systems

SK. Shaikh[1], Vasundha[2]
Department of Computer, Arts & Science College, Bhalod, M.S, India
Dept. of Computer, D.N. College, Faizpur, M.S, India

Research Article

Date of Submission: 10-01-2025 **Date of Acceptance: 12-04-2025** **Date of Publication: 19-05-2025**

Abstract—In this work, augmented Lagrangian-based techniques are developed and analyzed for two types of large-scale optimization problems that are pertinent to contemporary computing systems: the solution of regularized linear systems and energy-aware network routing with bandwidth allotment. In the first problem, a mixed-integer nonlinear algorithm is used to jointly optimize bandwidth allocation and routing in communication networks under energy restrictions. In the second, stability terms like Ψ2 regularization are introduced in regular-ized linear systems to solve ill-posed or ill-conditioned problems. For both problems, synchronous and asyn-chronous distributed optimization schemes are designed using problem characterized by discrete variables and inherent non-convexities. Such formulations pose significant computational challenges and demand algorithmic strategies that can scale while maintaining solution quality [3], [4].

In parallel, the solution of regularized linear systems has become increasingly important across domains such as signal processing, numerical optimization, and machine learning. Such problems arise when seeking stable solutions to ill-posed or ill-conditioned systems, where regularization terms (e.g., ℓ_2 penalties) are introduced to ensure robustness and decomposition techniques grounded in augmented Lagrangian theory. Extensive numerical experiments across diverse datasets, including network flow instances and benchmark regularized linear systems, demonstrate that the asynchronous variants retain comparable solution quality while significantly improving computational performance, particularly under delay and scala-bility conditions. These findings reinforce the practical value of asynchronous augmented Lagrangian methods for distributed, high-dimensional, and delay-sensitive optimization problems.

Keywords—Mixed-integer nonlinear programming (MINLP), dual decomposition, augmented Lagrangian techniques, Bertsekas decomposition, asynchronous and synchronous optimization, energy-conscious routing, bandwidth distribution, linear systems, distributed algorithms, and regularized linear systems

I. INTRODUCTION

The need for optimization frameworks that provide efficiency, scalability, and sustainability has never been greater due to the rise in large-scale data creation and real-time communication requirements. The simultaneous optimization of routing pathways and bandwidth allocations is crucial in modern telecommunication infrastructures to balance energy consumption and quality of service (QoS), two often conflicting goals. However, the possible performance is limited by the tendency of typical techniques to decouple these components or depend on static heuristics, especially when energy use is explicitly described as part of the system cost [1], [2]. A fully integrated approach yields a mixed-integer nonlinear programming (MINLP), which is more representative of real-world constraints.numerical stability [5], [6]. While these techniques enhance solution quality, they also introduce significant computational challenges, particularly in large-scale settings.

A unifying feature of these application domains is the re-liance on distributed optimization methods, which decompose large-scale problems into subproblems solvable by multiple agents in parallel. Classical synchronous approaches, such as the Bertsekas Augmented Lagrangian (BAL) framework, enforce strict coordination among agents, ensuring orderly updates but often at the cost of idle waiting and vulnerability to communication delays. Asynchronous methods [7]–[12], by contrast, allow agents to proceed with updates based on local and potentially outdated information, reducing synchronization overheads and improving resilience in large and heterogeneous networks. This paradigm shift has been shown to accelerate convergence in practice while maintaining theoretical guaran-tees under mild assumptions.

Building upon this perspective, the present work intro-duces asynchronous augmented Lagrangian schemes for two structurally analogous problems: energy-aware routing with bandwidth allocation and the solution of regularized linear systems. For the network routing problem, we extend the prior synchronous formulation [13] by incorporating asynchronous update mechanisms into a model that couples binary routing decisions with continuous flow allocations. For regularized linear systems, we develop a tailored Bertsekas Augmented Lagrangian scheme under inequality constraints, consider-ing both synchronized and asynchronized update schemes. Through comprehensive numerical experiments on synthetic and real-world datasets, we demonstrate that asynchronous implementations retain comparable solution accuracy and fea-sibility while yielding substantial reductions in runtime. The results confirm the robustness and versatility of Bertsekas-style

augmented Lagrangian decomposition for combinatorial and continuous optimization problems in distributed and delay-prone environments.

II. Network Optimization Problem of Simultaneous Routing and Bandwidth Allocation in Energy-Aware Networks

We can describe the problem of optimizing routing and flow rates simultaneously as identifying flow rates and routes (single paths) that satisfy network constraints for all possible source-destination pairs at the minimal cost. The variables constraint in (3) and three inequalities: (8), (6), and (7). The constraint (8) ensures single-path routing, while (7) enforces the relationship between the auxiliary and binary variables.

To solve the problem (1)-(9), we decompose it with respect to the flows $w \quad W$. For this purpose, we introduce the admissible set XBY_w for each flow w, which results from the constraints (3), (4), (6)-(9). This set for flow w is defined as *Network Optimization Problem*

1) Dataset Description: The algorithms developed in this

$$\sum_{k'=k-k_o}^{k} \quad \sum_{k'=k-k_o}$$

$$a_j(k', k) = 1, \quad a_j(k', k) \geq 0, \quad \forall k, j \in P \quad (55)$$

$$b_i(k', k) = 1, \quad b_i(k', k) \geq 0, \quad \forall k, i \in M \quad (56)$$

III. Numerical Tests

study were implemented and evaluated on three network topologies of increasing complexity: medium, large, and extra-large. Each topology was tested under multiple objective parameter configurations (γ, δ), yielding ten problem in-stances. The networks were designed as loosely connected clusters of nodes exhibiting strong intra-cluster connectivity. Each instance was solved using a synchronous and an asyn-chronous formulation to evaluate comparative performance.

A. Implementation Details

All implementations and numerical experiments were con-ducted using Python 3.12.0. The optimization models were formulated with the Pyomo modeling framework and solved using the Gurobi optimizer. Computational experiments were performed on a machine equipped with an AMD Ryzen 5 4600H processor (3.00 GHz, Radeon Graphics), 32 GB of RAM, and a 512 GB SSD, running a 64-bit Windows 10 Pro operating system. Network construction and visualization tasks were handled using the NetworkX library [15], facilitating the representation of graph-based structures integral to the optimization problems.

The individual results were first aligned to a unified time axis to enable consistent comparison across datasets with potentially different time indices. A comprehensive timeline was constructed by taking the union of all time points in the datasets. Each dataset was then merged onto this unified All experiments were executed in parallel using Python's `multiprocessing.Pool` to simulate distributed compu-tation. The detailed characteristics of the tested network in-stances are summarized in Table II.

2) Results and Discussion: In Table III, the objective values and runtimes are reported for each network problem instance under synchronous and asynchronous optimization, evaluated with and without simulated delays. Consistent trends are observed across all ten instances, spanning medium, large, and extra-large network topologies, which highlight the trade-offs between the two algorithmic paradigms. They can be visualize in Fig 1, Fig 2, Fig 3, Fig 4 and Fig 5.

Both formulations yielded nearly identical objective values across all settings without delay, indicating that the asyn-chronous method achieves comparable solution quality to the synchronous baseline. The minor discrepancies observed (e.g., Medium $\gamma = 2, \delta = 1$ with 163.49 vs. 163.54) are

TABLE II: Summary of network problem instances tested

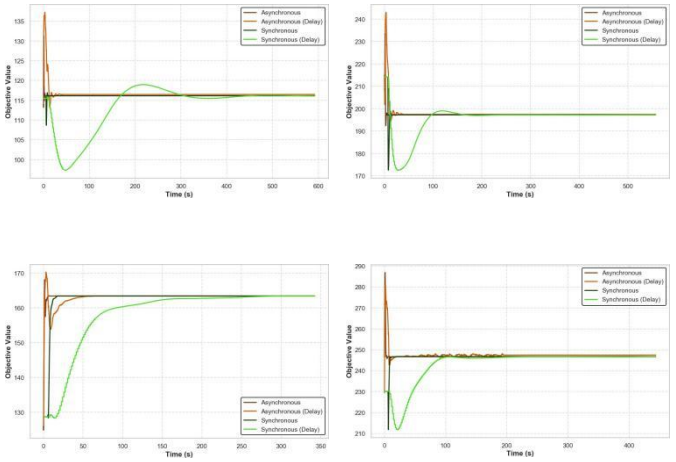
Problem Type	Number	Number	Number	Bounds	Bounds
Medium	25	84	12	[0·001, 3]	[0·3, 1]
Large	49	143	32	[0·001, 3]	[0·3, 1]
Extra Large	77	227	64	[0·001, 3]	[0·3, 1]

negligible and within numerical tolerance, affirming the cor-rectness of the asynchronous approach. Runtime comparisons without delay show a more nuanced pattern. In medium-sized networks, asynchronous optimization is generally completed faster than synchronous optimization. For instance, in the Medium $\gamma = 2, \delta = 1$ case, asynchronous execution completed in 10 seconds compared to 19 seconds for its synchronous counterpart. However, in some large network instances (e.g., Large $\gamma = 1, \delta = 2$), the asynchronous method was slightly slower (91 seconds vs. 58 seconds), likely due to the overhead of managing loosely coupled updates at scale. Nevertheless, the differences in runtimes were not drastic under normal execution conditions.

The introduction of artificial delays in selected data parti-tions dramatically magnified performance differences. In all medium and large network instances, synchronous runtimes increased significantly, often by an order of magnitude or more. For example, when delays were introduced, the syn-chronous runtime in the Medium $\gamma = 1, \delta = 1$ case jumped from 16 to 592 seconds. In contrast, the asynchronous runtime increased modestly, from 15 to 42 seconds. Similar resilience was observed across all configurations. Even in large-scale instances, the asynchronous approach maintained substantially lower runtimes than its synchronous counterpart under delayed conditions. For example, in the Large $\gamma = 1, \delta = 2$ case, the asynchronous runtime with delay was 183 seconds, while the synchronous method required 473 seconds.

Figure 5 presents a visualization of routing paths over a directed network graph (Small Problem Networks), where each node represents a router or a network element, and directed edges represent potential communication links. Using the spring layout for spatial arrangement, the plot shows the routing solutions for multiple commodities $w \quad W$, where each commodity represents a separate source-destination traf-fic demand pair. Flows are illustrated by color-coded directed edges, with each color corresponding to a different commodity. Edge labels indicate the commodity identifier and the flow value y_{wl} transmitted over that edge, truncated to a single decimal for clarity. Only edges with a non-negligible flow (above a threshold of 10^{-3}) are visualized to reduce clutter and emphasize active routes. Optionally, each commodity's source and destination nodes can be annotated with $S(w)$ and $D(w)$, respectively, to highlight the entry and exit points of data within the network. This visualization effectively captures both the structure of the network and the routing decisions for multiple commodities, enabling a direct comparison of path overlaps, bottlenecks, and routing efficiency across different scenarios or algorithmic configurations.

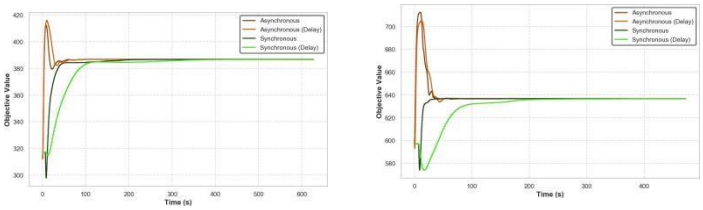
Overall, the results demonstrate that while both methods yield solutions of equivalent quality, asynchronous optimization exhibits significantly better robustness to delays and scales more gracefully with problem size; this makes it a compelling strategy for real-world distributed optimization problems where partial staleness and communication latency are common.



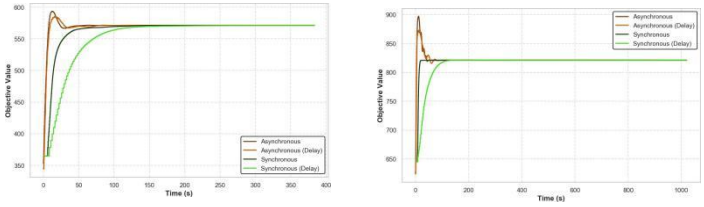
(a) $\gamma = 1, \delta = 1$ (b) $\gamma = 1, \delta = 2$

(c) $\gamma = 2, \delta = 1$ (d) $\gamma = 2, \delta = 2$

Fig. 1: Medium Problem Objective Values (Routing)



(a) $\gamma = 1, \delta = 1$ (b) $\gamma = 1, \delta = 2$



(c) $\gamma = 2, \delta = 1$ (d) $\gamma = 2, \delta = 2$

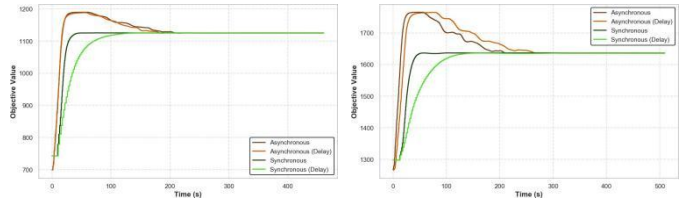
Fig. 2: Large Problem Objective Values (Routing)

C. Regularized Linear Systems

1) Dataset Description: The experiments are conducted on three linear systems designed to reflect varying data structures and matrix conditions. Two are derived from an image inpaint-ing task using randomized diagonal measurement operators,

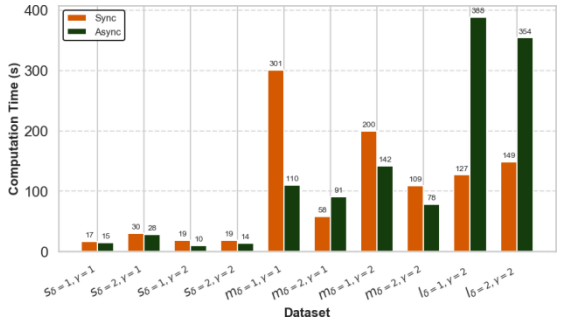
TABLE III: Routing Results

Data	Objective		Objective (with delay)		Time [s]		Time (with delay) [s]	
	Sync	Async	Sync	Async	Sync	Async		
Medium $\gamma = 1, \delta = 1$	116.13	116.13	116.1	116.52	16	15	592	42
Medium $\gamma = 1, \delta = 2$	197.27	197.27	197.27	197.47	30	28	557	38
Medium $\gamma = 2, \delta = 1$	163.49	163.54	163.49	163.54	19	10	342	65
Medium $\gamma = 2, \delta = 2$	246.84	246.69	246.85	246.62	19	14	443	195
Large $\gamma = 1, \delta = 1$	386.81	386.76	386.81	386.64	301	110	628	123
Large $\gamma = 1, \delta = 2$	636.63	636.78	636.63	636.68	58	91	473	183
Large $\gamma = 2, \delta = 1$	571.37	571.50	571.37	571.52	200	142	384	117
Large $\gamma = 2, \delta = 2$	821	821	821	821	109	78	1020	115
Extra Large $\gamma = 2, \delta = 1$	1125.21	1125.22	1125.21	1125.26	127	388	461	429
Extra Large $\gamma = 2, \delta = 2$	1636.44	1635.70	1636.44	1636.45	149	354	509	400

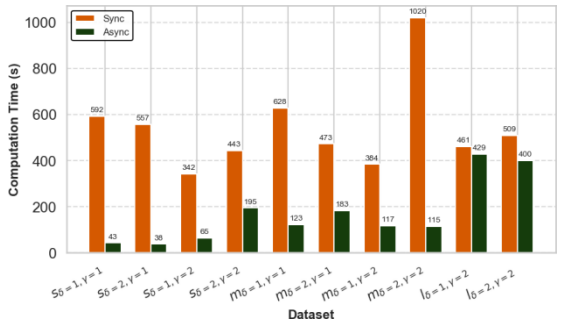


(a) $\gamma = 1, \delta = 1$ (b) $\gamma = 1, \delta = 2$

Fig. 3: Extra Large Problem Objective Values (Routing)



(a) Without Delay

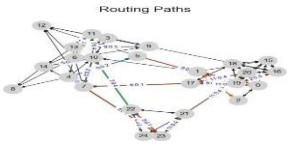


(b) With Delay

Fig. 4: Run Time (Routing)

and the third is based on a biomedical dataset reformulated as a compressed feature recovery problem. Each dataset adheres to the standard linear model $Ax = y$, with a known ground truth $x^\star$.

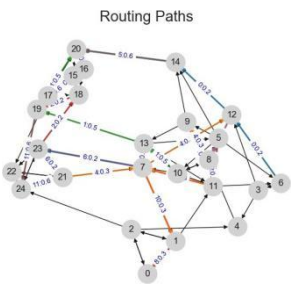
The first dataset simulates a diagonal sensing problem in grayscale image recovery. A 16×16 grayscale image is vectorized into $x^\star \in \mathsf{R}^{256}$, and a diagonal measurement matrix $A \in \mathsf{R}^{256\times256}$ is constructed with entries drawn

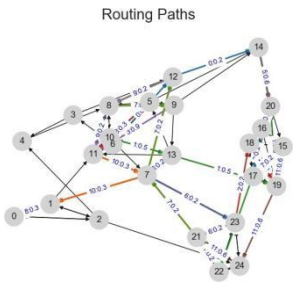


(a) Sync $\gamma = 1, \delta = 1$



(b) Async $\gamma = 1, \delta = 1$





(c) Sync $\gamma = 2, \delta = 2$ (d) Async $\gamma = 2, \delta = 2$

Fig. 5: Small Problem Networks (Routing)

independently from a uniform distribution over $[0, 1)$. The corresponding observation vector is computed as $y = Ax^\star$, resulting in a well-scaled and positive semi-definite system. This formulation captures moderate conditioning and serves as a stable reference for evaluating solution quality.

The second dataset uses the same underlying image and construction but replaces the uniform distribution with a stan-dard Gaussian distribution for the diagonal entries of A. This results in a more ill-conditioned system where entries can be both positive and negative, potentially with large magnitudes. The measurement vector $y = Ax^\star$, therefore, reflects a noisier and more variable scaling of the original image, posing greater challenges for algorithmic recovery under unstable and zero-mean multiplicative transformations.

The third dataset is derived from the UCI Breast Cancer Wisconsin (Diagnostic) dataset [17] and formulated as a compressed sensing task. After standardizing the data, a single feature vector $x^\star \in \mathbb{R}^{30}$ is selected from the training partition. A sensing matrix $A \in \mathbb{R}^{10\times 30}$ is generated with independent tandard Gaussian entries scaled by $1/\sqrt{10}$. The measurement vector $y = Ax^\star$ thus represents a low-dimensional projection of the original biomedical profile. This setting is representative of practical dimensionality reduction problems in clinical data, where reconstruction must be achieved from limited and noisy observations.

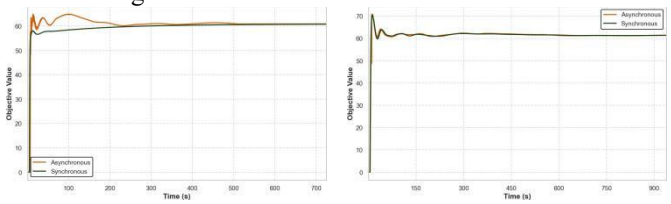
2) Results and Discussion: Table IV, Fig 6, Fig 8, Fig 9, Fig 10, and Fig 7 report the performance of the synchronous and asynchronous variants of the Bertsekas algorithm on three regularized linear systems: one derived from clinical data (Cancer) and two from synthetic image recovery tasks with uniform and Gaussian diagonal measurement matrices. All experiments were conducted using 12 data partitions to reflect realistic distributed processing conditions.

Across all three datasets, both synchronous and asyn-chronous algorithms achieved the same objective values, indi-cating convergence to consistent solutions. For the synthetic image-based systems, the objective values were 60.87 and 61.41 for the uniform and Gaussian matrices, respectively. The Cancer dataset yielded a lower final objective value of 54.61, likely due to its lower dimensionality and better numerical conditioning compared to the randomized sensing matrices used in the synthetic cases. The consistency in solution quality across all runs confirms that asynchronous updates do not compromise the convergence accuracy of the algorithm, even in ill-conditioned or noisy settings.

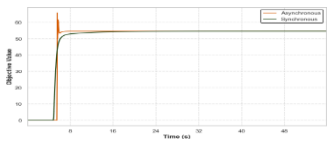
In terms of runtime, the asynchronous variant demonstrated clear computational advantages. On the low-dimensional Can-cer dataset, it reduced the execution time from 55 seconds (synchronous) to just 8 seconds, yielding a nearly seven-fold speedup. This is particularly relevant in real-world scenar-ios involving clinical diagnostics, where quick and reliable inference is essential. For the Gaussian diagonal sensing matrix, the asynchronous version reduced runtime from 724 to 511 seconds—approximately a 30% improvement. While the gain was less dramatic in the uniform matrix case (938 vs. 918 seconds), the asynchronous approach still maintained its efficiency without introducing instability.

Overall, all configurations resulted in successful conver-gence. The asynchronous scheme consistently matched the solution quality of its synchronous counterpart while sig-nificantly reducing computation time in most cases. These results suggest that asynchronous distributed optimization is especially effective in scenarios involving low-dimensional or poorly conditioned systems, where it can deliver substantial efficiency gains without sacrificing solution accuracy.

TABLE IV: Performance Comparison of Optimization Algo-rithms for Regularized Systems of Linear Equations Across Different Problem Regimes with 12 Partitions



(a) Uniform (b) Gaussian



(c) Cancer Spars

Fig. 6: Async-Sync: Objectives

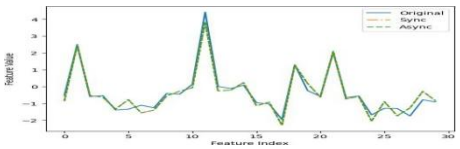


Fig. 7: Async-Sync: Compressed Measurements of Cancer Spars

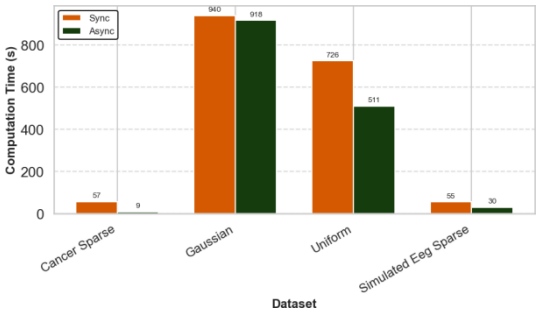
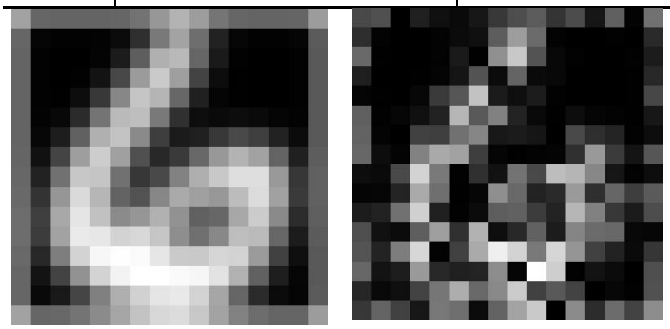
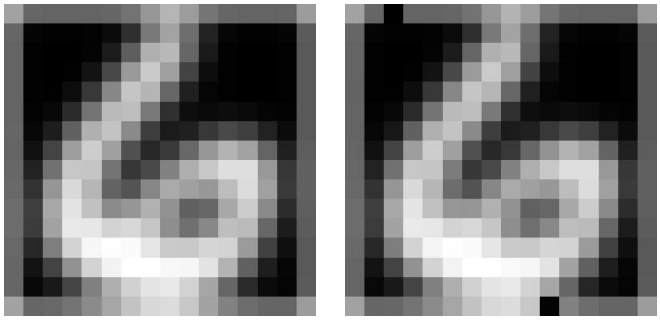


Fig. 8: Async-Sync: Run Time (secs)

Dataset	Synchronous Objective	Synchronous Time(s)	Synchronous Status	Asynchronous Objective	Asynchronous Time(s)	Asynchronous Status
Gaussian	61.41	724	Converged	61.41	511	Converged
Uniform	60.87	938	Converged	60.87	918	Converged
Cancer	54.61	55	Converged	54.61	8	Converged



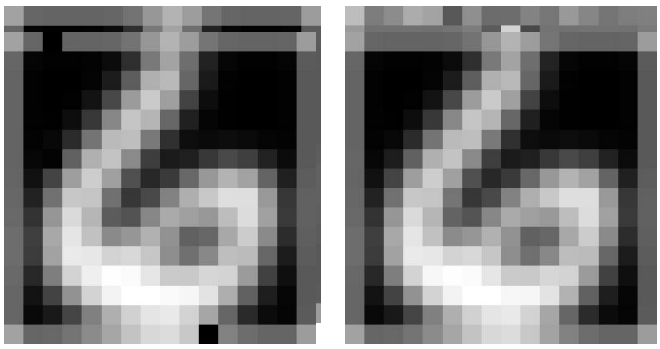
(a) Original (b) Inpainted (Noisy)



(c) Synchronous (d) Asynchronous

Fig. 9: Image Recovery under Gaussian Additive Noise

(a) Original (b) Inpainted (Noisy)



(c) Synchronous (d) Asynchronous

Fig. 10: Image Recovery under Gaussian Additive Noise

Conclusions and Future Work

In order to solve regularized linear systems in large-scale machine learning contexts and optimize routing and bandwidth allocation simultaneously in energy-aware networks, this study developed and tested Bersekas Augmented Lagrangian decomposition algorithms (Synchronous and Asynchronous). The difficulties caused by the large dimensionality of the regularized linear systems and the mixed-integer character of routing issues were solved by the suggested methods. The developed techniques used dualization and decomposition techniques to produce scalable and effective computing. Notably, asynchronous systems outperformed their synchronous counterparts in terms of computing performance and tolerance to delays, all without appreciably degrading solution quality.

The resilience and flexibility of asynchronous optimization paradigms in large-scale and distributed systems are shown by this study, which goes beyond empirical validation. Our approach overcomes the conventional drawbacks of Lagrangian relaxation in the context of network design by generating workable, superior routing and flow allocations. Asynchronous update techniques provide significant wall-clock time savings in regularized linear systems, particularly in heterogeneous computing environments.

In the future, this paradigm creates a number of opportunities for further study. One approach is to include stochastic or uncertain demand profiles, which represent more realistic operational circumstances, and expand the network model to enable multi-path routing. Improving the convergence theory of asynchronous techniques, particularly under loosened assumptions like unbounded delays or non-uniform update frequency, is another potential area of research.

Adaptive step-size techniques and dynamic penalty parameter adjustment may also speed up convergence. Lastly, implementation on distributed edge platforms or actual networking gear would provide useful information on the practicality of these techniques in real-time applications. In networked and data-rich situations, these paths seek to further the use of augmented Lagrangian techniques as a basis for effective, scalable, and decentralized optimization.

References

[1] E. Gelenbe, “Energy packet networks: adaptive energy management for the cloud,” in *Proceedings of the 2nd International Workshop on Cloud Computing Platforms*, ser. EuroSys ’12. ACM, apr 2012, p. 1–5. [Online]. Available: http://dx.doi.org/10.1145/2168697.2168698

[2] B. Wang and P.-H. Ho, “Energy-efficient routing and bandwidth allocation in ofdm-based optical networks,” *Journal of Optical Communications and Networking*, vol. 8, no. 2, p. 71, jan 2016. [Online]. Available: http://dx.doi.org/10.1364/JOCN.8.000071

[3] S. Burer and A. N. Letchford, “Non-convex mixed-integer nonlinear programming: A survey,” *Surveys in Operations Research and Management Science*, vol. 17, no. 2, p. 97–106, 7 2012. [Online]. Available: http://dx.doi.org/10.1016/j.sorms.2012.08.001

[4] I. E. Grossmann and Z. Kravanja, *Mixed-Integer Nonlinear Programming: A Survey of Algorithms and Applications*. Springer New York, 1997, p. 73–100. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-1960-6_5

[5] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. New York: Wiley, 1977.

[6] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numer-ical Aspects of Linear Inversion*. Philadelphia: SIAM, 1998.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computa-tion: Numerical Methods*. Prentice Hall Inc., 1989.

[8] ——, “Some aspects of parallel and distributed algorithms: A survey,” *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

[9] S. Low and D. Lapsely, “Optimization flow control. i. basic algorithm and convergence,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, p. 861–874, 1999. [Online]. Available: http://dx.doi.org/10.1109/90.811451

[10] A. Karbowski, “Comments on “optimization flow control .i. basic algorithm and convergence”,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, p. 338–339, 4 2003. [Online]. Available: http://dx.doi.org/10.1109/TNET.2003.810318

[11] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed admm for large-scale optimization—part i: Algorithm and convergence analysis," in *IEEE Transactions on Signal Processing*, vol. 64, 2016, pp. 118–132.E. Wei and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers," 2013. [Online]. Available: https://arxiv.org/abs/1307.8254

[12] A. C. Nwachukwu and A. Karbowski, "Solution of the simultaneous routing and bandwidth allocation problem in energy-aware networks using augmented lagrangian-based algorithms and decomposition," *Energies*, vol. 17, no. 5, p. 1233, 3 2024. [Online]. Available: http://dx.doi.org/10.3390/en17051233

[13] I. Ruksha and A. Karbowski, "Decomposition methods for the network optimization problem of simultaneous routing and bandwidth allocation based on lagrangian relaxation," *Energies*, vol. 15, no. 20, p. 7634, 10 2022. [Online]. Available: https://doi.org/10.3390/en15207634

[14] "Networkx - network analysis in python," https://networkx.org/, ac-cessed: 2024-02-12.

[15] The Pandas Development Team, "pandas.merge_asof — pandas documentation," 2024, accessed: 2025-06-20. [Online]. Available: https://pandas.pydata.org/docs/reference/api/pandas.merge_asof.html

[16] W. H. Wolberg and O. L. Mangasarian, "Breast cancer wisconsin (diagnostic)," 1993.