Constrained LMS for Dynamic Flow Networks

Konstantinos Eftaxias and Clive Cheong Took
Department of Computer Science
University of Surrey
Guildford, UK
{k.eftaxias,c.cheongtook}@surrey.ac.uk

Bruno Venturini
WSP|PB UK
Basingstoke, UK
bruno.venturini@wspgroup.com

David Arscott
PyTerra Ltd
Godalming, UK
d.arscott@pyterra.co.uk

***Abstract*—In this era of climate change, there is a growing need to offer adaptive learning algorithms in the optimisation of natural resources. These resources are typically optimised by evolutionary algorithms. However, evolutionary algorithms (EAs) are no longer adequate due to the 'drift' component introduced by environmental factors such as flash flooding. We therefore propose a novel constrained Least Mean Squares (LMS) algorithm for the optimisation of flow networks. For rigor, we provide a stability analysis of our adaptive algorithm, which enables us to interpret the physical meaning of the network at equilibrium. We evaluate our proposed method against genetic algorithm (GA), the most common evolutionary algorithm. The results are promising: not only the proposed constrained LMS has a performance advantage over GA, but its computational cost is significantly lower making it more suitable for real-time applications.**

1. Introduction

The need for smart systems for the management of our natural resources has become essential for the sustainability of the environment. Moreover, unpredictable weather events can create significant problems and planning ahead is essential for the minimisation of the impact. The nonstationarity of environmental data means that any machine learning solution should be able to cater for the 'drift' and therefore be adaptive in nature. In this paper, we propose a smart system for the optimisation of flow networks in the management of water resources.

In the past decade, there has been a strong emphasis in the computational intelligence community to address smart grid applications, yet other network-based problems such as those in flow networks (e.g. water networks) have not received as much attention [1]. As a result, algorithmic solutions for water and gas networks are still at an early stage compared to their smart grid counterpart. One of challenges encountered in flow networks lies in the limited level of actuation to automate such networks. However, with advent of Internet of Things, the increasing ubiquity of sensors and actuators in smart city applications first highlights that smart automation of flow networks is not fiction, and second smart systems are becoming increasingly important, especially in this era of climate change [2][3]. To cater for such nonstationary conditions, we will focus on the adaptive optimisation of flow networks.

Flow networks are defined as directed graphs with specific edge capacity and can model accurately gas and water networks. Previous works have focused mainly on the exploitation of evolutionary algorithms (EAs) for the long-term planning and management of natural resources. In this context, EAs (e.g. genetic algorithm) were optimal in the design and rehabilitation of gas and water networks [4]–[7]. However, EAs are non-adaptive. As such, non-stationary conditions cannot be naturally accounted for, especially in the context of dynamic environmental changes [8]. Our work addresses this shortcoming in the literature to propose an adaptive solution, which can cope with real-time applications. We are focusing on the resource management problem on a dynamic environment which is essential for an optimised smart network.

Specifically, we propose a constrained least mean squares (LMS) based algorithm that can simulate and optimise accurately the operation of a smart network. Each node of the network has a specific target in terms of volume and the algorithm control the flows in order the target to be met. The targets are set based on various factors, such as consumption demand, weather forecasting and environmental aspects. The adaptive properties of the method ensures that even when the targets are dynamically changing the algorithm always iterates to find the optimum solution. Especially, in extreme conditions (for example, a flash flood), the responsiveness of such an algorithm is a high priority. In Section 2, we introduce the constrained LMS for flow networks and illuminate the optimal conditions of our constrained LMS algorithm at convergence. In Section 3, we evaluate our proposed method against genetic algorithm (GA), which is the established method in optimising flow networks [4][9]. Finally, we conclude by assessing the results and highlight the advantages of our method.

In this paper, we use normal font for scalars. To denote matrices we use bold capital letters while for vectors, we use bold lower-case letters. All vectors are column vectors. The transpose of a matrix or a vector is denoted by the superscript $(\cdot)^T$ and $\|\cdot\|$ represents the ℓ_2-norm.

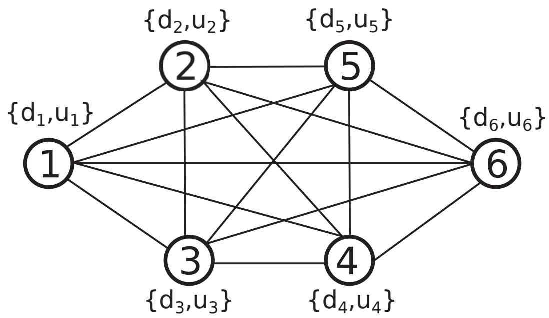

Figure 1: A fully connected flow network with $N = 6$ nodes.

2. Constrained LMS for flow networks

Consider a network of N nodes as in Fig. 1. Each node has a value $u_k(t)$ which represents the capacity of each node. Moreover, there is a target value $d_k(t)$ that represents the desired amount to be stored within the kth node. If we write the same values in vector form $\{\mathbf{d}_t, \mathbf{u}_t\}$ collecting all the values from the network, we can introduce an $N \times N$ matrix $\mathbf{A}$ that represents the flows between the nodes. The amount of material that flows from node j to node i is expressed as a percentage of $u_j(t)$ in terms of α_{ij} of the matrix $\mathbf{A}$. Due to the non-negative nature of the quantity 'percentage', the first constraint is to enforce non-negativity on the matrix $\mathbf{A}$. The 'flow' of the network can now be modelled as

$$\mathbf{u}_{t+1} = \mathbf{A}\mathbf{u}_t \tag{1}$$

The main objective is to estimate the optimal value of $\mathbf{A}$ such that $\mathbb{E}\,\|\mathbf{d} - \mathbf{A}\mathbf{u}\|^2$ is minimised (where $\mathbb{E}$ denotes the expectation operator). Thus, the cost function of the optimisation problem can be formulated as:

$$J(\mathbf{A}) = \mathbb{E}\,\|\mathbf{d} - \mathbf{A}\mathbf{u}\|^2 \tag{2}$$

Moreover, we need to enforce the second constraint, which is the conservation of mass for the whole system. Therefore, each column of the matrix $\mathbf{A}$ should sum to 1. This can be written in a compact form as:

$$\mathbb{1}_N^T \mathbf{A} = \mathbb{1}_N^T \tag{3}$$

where $\mathbb{1}_N$ is a $N \times 1$ vector of ones. Thus, the Lagrangian can be written as:

$$L(\mathbf{A}, \boldsymbol{\lambda}) = \mathbb{E}\,\|\mathbf{d} - \mathbf{A}\mathbf{u}\|^2 + (\mathbb{1}_N^T \mathbf{A} - \mathbb{1}_N^T)\boldsymbol{\lambda} \tag{4}$$

where $\boldsymbol{\lambda}$ denotes the vector of the Lagrange multipliers. The Karush-Kuhn-Tucker conditions must be satisfied at the optimum solution $\{\mathbf{A}^o, \boldsymbol{\lambda}^o\}$, meaning that

$$\nabla_{\mathbf{A}} L(\mathbf{A}^o, \boldsymbol{\lambda}^o) = 0 \tag{5}$$

where $\nabla_{\mathbf{A}}$ is the gradient with respect to $\mathbf{A}$. Substituting the Lagrangian of Eq. (4) into Eq. (5) we obtain the following equation:

$$2(\mathbf{d} - \mathbf{A}\mathbf{u})\mathbf{u}^T + \mathbb{1}\boldsymbol{\lambda}^T = 0 \tag{6}$$

$$\mathbb{1}\boldsymbol{\lambda}^T = -2(\mathbf{d} - \mathbf{A}\mathbf{u})\mathbf{u}^T \tag{7}$$

In order to replace the Lagrange multipliers in Eq. (4), we need to estimate the terms $\mathbb{1}_N^T \boldsymbol{\lambda}$ and $\mathbb{1}_N^T \mathbf{A}\boldsymbol{\lambda}$. Using Eq. (7) and replacing the terms $\mathbf{u}\mathbf{u}^T$ and $\mathbf{d}\mathbf{u}^T$ with the covariance $\mathbf{R}_u$ and cross covariance $\mathbf{R}_{du}$ matrices respectively, we can express the terms of Eq. (4) as:

$$\begin{aligned} \mathbb{1}_N^T \boldsymbol{\lambda} &= \mathrm{Tr}(\mathbb{1}_N \boldsymbol{\lambda}^T) \\ &= 2\,\mathrm{Tr}(\mathbf{A}\mathbf{R}_u) - 2\,\mathrm{Tr}(\mathbf{R}_{du}) \end{aligned} \tag{8}$$

$$\begin{aligned} \mathbb{1}_N^T \mathbf{A}\boldsymbol{\lambda} &= \mathrm{Tr}(\mathbb{1}_N \boldsymbol{\lambda}^T \mathbf{A}) \\ &= 2\,\mathrm{Tr}(\mathbf{A}\mathbf{R}_u\mathbf{A}) - 2\,\mathrm{Tr}(\mathbf{R}_{du}\mathbf{A}) \end{aligned} \tag{9}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace operator. Substituting the above expressions into Eq. (4), the Lagrangian function can conveniently be expressed as:

$$\begin{aligned} L(\mathbf{A}) = {} & \mathbb{E}\,\|\mathbf{d} - \mathbf{A}\mathbf{u}\|^2 + 2\,\mathrm{Tr}(\mathbf{A}\mathbf{R}_u\mathbf{A}) \\ & - 2\,\mathrm{Tr}(\mathbf{R}_{du}\mathbf{A}) - 2\,\mathrm{Tr}(\mathbf{A}\mathbf{R}_u) + 2\,\mathrm{Tr}(\mathbf{R}_{du}) \end{aligned} \tag{10}$$

The condition of Eq. (5) needs to be satisfied in this new form of the Lagrangian, that is,

$$\nabla_{\mathbf{A}} L(\mathbf{A}^o) = 0. \tag{11}$$

By calculating the gradient of the Lagrangian expression in Eq. (10) with respect to $\mathbf{A}$, we reach the following conditions:

$$2(\mathbf{d} - \mathbf{A}\mathbf{u})\mathbf{u}^T - 2\mathbf{R}_{du}^T + 2\mathbf{A}^T\mathbf{R}_u^T + 2\mathbf{R}_u^T\mathbf{A}^T - 2\mathbf{R}_u^T = 0$$

$$\underbrace{(\mathbf{R}_{du} - \mathbf{R}_{du}^T)}_{\text{Term I}} + \underbrace{(\mathbf{A}^T - \mathbf{A})}_{\text{Term II}}\mathbf{R}_u + \mathbf{R}_u\underbrace{(\mathbf{A}^T - \mathbf{I}_N)}_{\text{Term III}} = 0 \tag{12}$$

where $\mathbf{I}_N$ denotes the $N \times N$ identity matrix.

If we analyse Equation (12), which reflects the convergence of the proposed constrained LMS, the terms (I)-(III) can be interpreted as the conditions for stability at convergence:

$$\text{I. } \mathbf{R}_{du} - \mathbf{R}_{du}^T = 0 \tag{13}$$

$$\text{II. } \mathbf{A}^T - \mathbf{A} = 0 \tag{14}$$

$$\text{III. } \mathbf{A}^T - \mathbf{I} = 0. \tag{15}$$

Condition (I) implies that the cross-covariance between the target $\mathbf{d}$ and the current $\mathbf{u}$ volume of each node must be symmetric, however cross-covariance is generally non-symmetric. The symmetricity of the cross covariance is due to the approximation $\mathbf{d} \approx \mathbf{A}\mathbf{u}$ from (2) at convergence. In other words, the current volume of each node has reached the target volume at convergence. This is further confirmed by Condition (III), which implies that $\mathbf{A} = \mathbf{I}$.

Condition (II) means the flow matrix $\mathbf{A}$ needs to be symmetric. In fact, Condition (II) conforms with Condition (III) that $\mathbf{A} = \mathbf{I}$, as Condition (III) means that there is no flow between the nodes (α_{ij}=$\alpha_{ji} = 0$). Moreover, Condition (III) verifies the mass conservation constraint of Eq. (3) still holds at convergence. Now that we have analysed the constrained LMS algorithm, we proceed on the simulation section to evaluate our proposed algorithm.

Option	Method
Creation	Uniform Distribution
Selection	Stochastic Uniform Function
Mutation	Gaussian Function
Crossover	Scatter Function

TABLE 1: Genetic algorithm options

Experiment	Constr. LMS	GA
Single Target	139.1	344.5
Triple Target	315.8	606.4

TABLE 2: Method Comparison based on total error

3. Simulations

This section provides two sets of experiments simulating a water supply network. The first experiment simulated a non-drift environment, whereas the second one simulated a drift environment. In these two sets of simulations, we evaluated the performance of constrained LMS against the performance of a genetic algorithm [9].

Consider a water network of $N = 6$ nodes as shown in Fig. 1. Each node had a specific target volume value d_k and a initial volume value $u_k(0)$ assigned to it. The initial and target values were taken from a random number generator with a uniform distribution and a range from 0 to 100 m^3. For these two sets of simulation, the constrained LMS algorithm was updated as follows:

$$\mathbf{A}_{t+1} = \mathbf{A}_t + \mu(\mathbf{d}_t - \mathbf{A}_t\mathbf{u}_t)\mathbf{u}_t^T \quad (16)$$

After each update step, a set of constraints were enforced to the estimate $\mathbf{A}$ in order to keep the solution conformed with the physical properties of the system. Specifically, as the elements α_{ij} of the flow matrix $\mathbf{A}$ represent volume percentages, these quantities needed to be nonnegative. Therefore, the first constraint was expressed as:

$$\alpha_{ij} \geq 0, \ \ \forall i, j \quad (17)$$

The second constraint was based on the flow limits we have on physical water networks between the nodes. In order to achieve a realistic simulation of a water network, we set a maximum flow value between the nodes. As such, we used a 5% upper limit for every connection within the network. The only matrix elements that did not follow this constraint were the diagonal elements of $\mathbf{A}$. Its diagonal elements reflected the amount of water that was kept at a specific node after each iteration. The second constraint can therefore be expressed as:

$$\alpha_{ij} \leq 0.05, \ \ \forall i, j \text{ with } i \neq j \quad (18)$$

Hence, the values of α_{ij} were bounded by the lower (17) and then upper (18) limit. Finally, the third constraint (3) on the mass conservation was also enforced at each iteration of (16). The diagonal values were estimated in such a way that each column of $\mathbf{A}$ summed to unity. This way, we circumvented any normalisation step that would violate the first two constraints.

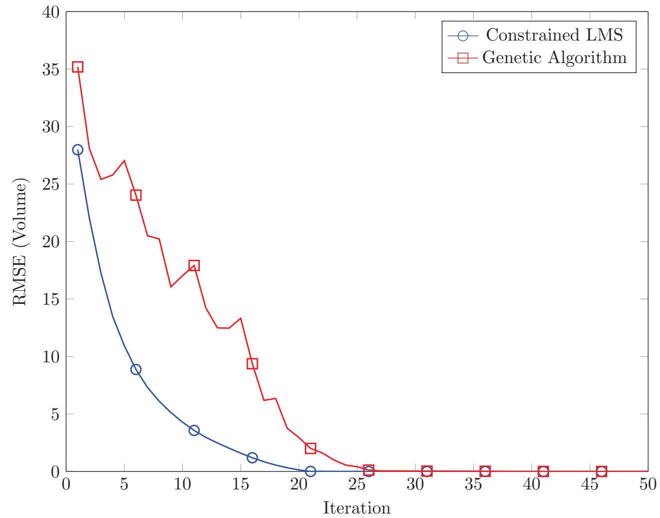

Figure 2: Learning curves for constrained LMS and GA. Constrained LMS performance is significantly better that the GA.

As for the genetic algorithm (GA), the same cost function (2) and set of constraints (3), (17)-(18) were used. Furthermore, we considered the GA options as shown in Table 1. At each iteration, GA found an estimation of $\mathbf{A}^o$, which was used a starting point for the next iteration. Therefore, there was a continuity in the search of the optimum solution.

On the first experiment the same targets and initial volumes were used for both methods. For constrained LMS, the learning rate μ was set at 4.5×10^{-4} in (16), a value that offered a good balance between convergence speed and stability. The performance metric used was the root mean squared error (RMSE) between the target and actual volumes. Mathematically, this metric can be computed as:

$$\text{RMSE}(t) = \sqrt{\mathbb{E}((\mathbf{d} - \mathbf{u}_t)^2)} \quad (19)$$

The results of the first experiment are shown in Fig. 2. It is clear from the figure that constrained LMS converged after 21 iterations while the GA converged after 26 iterations. In other words, the rate of convergence of the constrained LMS was greater than that of GA rate. This can be crucial for critical applications where the system needs to be highly reactive to rapid changes of the environment.

In the second experiment, the target volumes of each node were dynamically changed. In particular, we employed the same mechanism to set three different random sets of targets and assign them to three equal intervals (1-50, 51-100, 101-150). As shown in Fig. 3, constrained LMS performed better. In particular, the difference between the error for the two methods was significant. The adaptive properties of the LMS has proven to be valuable in applications where the data are dynamic.

In order to quantify the results, the sum of errors were computed and presented in Table 2 for both methods and experiments. Constrained LMS achieved nearly twice as better results than the GA method. On comparing the two methods, we can make the following remarks:

- Constrained LMS converged not only faster, but its convergence was smoother than the GA equivalent. In particular, GA's performance on the transient state was not as smooth as that of the LMS algorithm, which was reflected by the small peaks at Iteration 5 and 11 in Fig. 2. This is not surprising due to the meta-heuristic nature of EAs.
- At the steady state, the constrained LMS performed better than GA. The error values that were achieved from GA were of the order of 10^{-4} whereas for the LMS, the errors were of the order of 10^{-10} after 50 iterations.
- Consequently, the constrained LMS outperformed the GA equivalent not only at the transient state but also at the steady state. Table 2 confirms the overall performance superiority of our proposed method over GA.
- The computational cost of GA is significantly larger than the cost of the constrained LMS. Indeed, a simplified approximation of the time complexity of GA is of the order of $\mathcal{O}(gmN^2)$ per iteration where g is the number of generations, m is the size of the population and N the number of nodes. On the other hand, the time complexity of LMS is of the order of $\mathcal{O}(N^2)$ per iteration. As the size of the population and the number of generations is not fixed, it is difficult to make an exact estimation of the time complexity difference. However, it is clear from the above figures that LMS is much cheaper than the GA algorithm. On the actual running time of the experiment two, it took approximately 100 minutes for the GA to complete all the 150 iterations while the constrained LMS completed 150 iterations in less than 1 second.
- GA is not, by design, an adaptive algorithm, whereas constrained LMS is inherently adaptive. As such, we had to adapt GA by feeding in the estimate of the previous iteration as a starting point for the next iteration.
- The optimisation problem considered for both experiments were based on a single objective (2), however, if the optimisation problem considered was set as multi-objective one, it is envisaged that the genetic algorithm would outperform the constrained LMS. For such multi-objective problems [5], the genetic algorithm would converge but would not be able to cope with real-time applications, whereas the constrained LMS would not converge at all. To this end, we require a hybrid algorithm that exploits the global convergence property of an evolutionary algorithm and the lower computational cost and adaptive operation of the constrained LMS, which is the subject of future research.

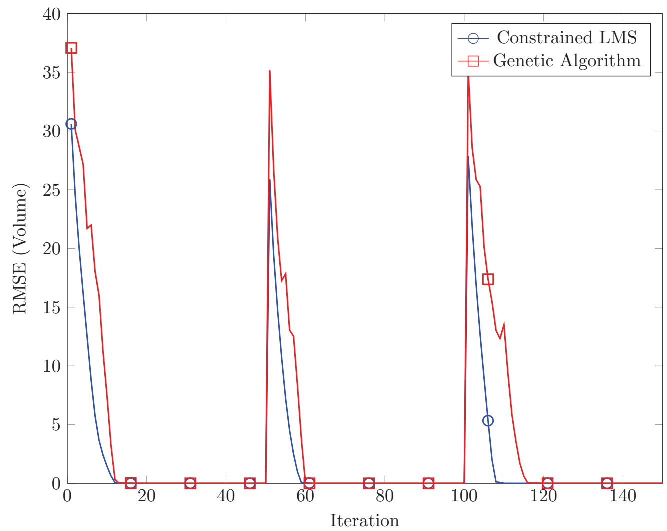

Figure 3: Learning curves for three changing targets.

4. Conclusion

We have proposed a constrained LMS algorithm to optimise flow networks. In particular, we have shown how the adaptive nature of the constrained LMS algorithm caters for the 'drift' based on varying targets. For rigour, we have also provided a stability analysis, which enabled us to interpret mathematically that at convergence, there is no flow between nodes and that each node has reached the desired target volume, as expected. As an example of a real-world application, we have considered a water supply network taking into consideration the physical constraints of the system and showed that our proposed LMS algorithm outperformed the genetic algorithm in simulations that accounted for both stationary and non-stationary environment. For single objective optimisations, our results indicate the suitability of our proposed LMS algorithm for real-time operation, which can be beneficial in this era of flash flooding.

References

[1] M. Fagiani, S. Squartini, L. Gabrielli, M. Pizzichini, and S. Spinsante, "Computational intelligence in smart water and gas grids: An up-to-date overview," *International Joint Conference on Neural Networks (IJCNN)*, pp. 921–926, 2014.

[2] G. Cembrano, G. Wells, J. Quevedo, R. Pérez, and R. Argelaguet, "Optimal control of a water distribution network in a supervisory control system," *Control engineering practice*, vol. 8, no. 10, pp. 1177–1188, 2000.

[3] P. Verma, A. Kumar, N. Rathod, P. Jain, S. Mallikarjun, R. Subramanian, B. Amrutur, M. Kumar, and R. Sundaresan, "Towards an iot based water management system for a campus," *IEEE First International Smart Cities Conference (ISC2)*, pp. 1–6, 2015.

[4] Z. Y. Wu and A. R. Simpson, "Competent genetic-evolutionary optimization of water distribution systems," *Journal of Computing in Civil Engineering*, vol. 15, no. 2, pp. 89–101, 2001.

[5] P. D. Surry, N. J. Radcliffe, and I. D. Boyd, "A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method," *AISB Workshop on Evolutionary Computing*, pp. 166–180, 1995.

[6] H. R. Maier, A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan, "Ant colony optimization for design of water distribution systems," *Journal of water resources planning and management*, vol. 129, no. 3, pp. 200–209, 2003.

[7] D. Halhal, G. A. Walters, D. Ouazar, and D. A. Savic, "Water network rehabilitation with structured messy genetic algorithm," *Journal of Water Resources Planning and Management*, vol. 123, no. 3, pp. 137–146, 1997.

[8] R. Wardlaw and M. Sharif, "Evaluation of genetic algorithms for optimal reservoir system operation," *Journal of water resources planning and management*, vol. 125, no. 1, pp. 25–33, 1999.

[9] S. Wan, Y. Hao, and C. Lv, "Research on water supply network optimization using advanced GA," *Second International Conference on Information and Computing Science*, vol. 1, pp. 243–246, 2009.