

A Clustering Approach to Edge Controller Placement in Software Defined Networks with Cost Balancing

Reza Soleymanifar, Amber Srivastava, Carolyn Beck, Srinivasa Salapaka

Abstract—In this work we introduce two novel deterministic annealing based clustering algorithms to address the problem of Edge Controller Placement (ECP) in wireless edge networks. These networks lie at the core of the fifth generation (5G) wireless systems and beyond. These algorithms, ECP-LL and ECP-LB, address the dominant leader-less and leader-based controller placement topologies and have linear computational complexity in terms of network size, maximum number of clusters and dimensionality of data. Each algorithm tries to place controllers close to edge node clusters and not far away from other controllers to maintain a reasonable balance between synchronization and delay costs. While the ECP problem can be conveniently expressed as a multi-objective mixed integer non-linear program (MINLP), our algorithms outperform state of art MINLP solver, BARON both in terms of accuracy and speed. Our proposed algorithms have the competitive edge of avoiding poor local minima through a Shannon entropy term in the clustering objective function. Most ECP algorithms are highly susceptible to poor local minima and greatly depend on initialization.

Keywords: Clustering, deterministic annealing, 5G networks, software defined networks, wireless edge networks, edge controller placement

I. INTRODUCTION

Wireless networks are of high importance in modern telecommunication systems as they are efficient, mobile, responsive, accessible, have enhanced guest access and better support expansion of network. In order to enhance these systems, Software-Defined Networks (SDN) have been introduced as an emerging paradigm whose primary advantage is giving developers greater control over the network traffic and administration [Alshamrani et al., 2018]. Traditionally wireless networks have played both the role of administration and relay of data within the same infrastructure. One of the limitations of this architecture is that modifying these networks requires manually re-configuring nodes of the network to accommodate the new changes. Softwarization is a new trend in wireless communication networks that helps to automate this type of manual work.

One of the most studied open research problems, on which SDN itself heavily relies, is the so-called edge controller placement problem (ECP) [Alshamrani et al., 2018]. Controller placement is one of the most important components of software defined networks [Kuang et al., 2018]. This problem was first introduced in [Heller et al., 2012] and is in general NP-hard [Singh and Srivastava, 2018]. Controllers are network nodes which are designated to control other nodes of a

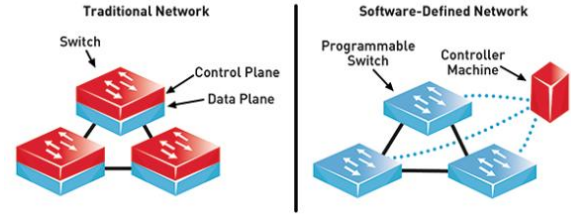


Fig. 1. Traditional versus Software Defined Networks. Image from [Dungay, 2016]

network. ECP in a fog/cloud network essentially reduces to determining how many and which nodes in the network need to be designated as the controllers. This placement induces several costs including delays between edge nodes and the controllers they are assigned to, and synchronization delay between the controllers themselves which we refer to as delay and synchronization costs respectively throughout the paper.

There are several approaches to address ECP problem. Our approach here is based on viewing this problem in a data clustering sense. Many clustering based approaches in literature are hindered by naive initialization and are thus prone to poor local optima. This leads to multiple optimization attempts with varied initializations that increase total computation time needed to find an optimal placement. These approaches are also restricted to a single objective value which prevents the decision maker from simultaneously considering multiple controller placement criteria. In this paper, we discuss the use of the deterministic annealing (DA) algorithm, which is tailored to avoid these shortcomings, and introduce algorithms that iteratively minimize the costs associated with ECP. In order to evaluate our algorithms we compare the final costs incurred with those of the MINLP formulation.

We identify the core competences of our algorithms as being (1) scalable and fast, due to linear computational complexity in terms of problem size and number of controllers, (2) high quality in terms of near optimal solutions, (3) initialization independent as we always start with one controller in the mass center of data, (4) excellent at avoiding

poor local minima due to the use of a Shannon entropy term in the clustering objective function and (5) able to address a multi-objective scheme.

The rest of the paper is organized as follows. In Section II we overview ECP and SDN related works from recent years. In Section III we concretely define ECP and explain the subtleties of this problem. In Section IV we describe our approach to the problem and explain how we adapt the DA to the ECP problem. The reader may refer to Section V to see the results of the simulations and finally Section VI shows conclusions and avenues for future research.

II. LITERATURE REVIEW

The controller placement problem for SDNs was first introduced in [Heller et al., 2012]. [Li and Xu, 2018] implement the Cuckoo search algorithm for the problem of controller placement in SDNs. [Lu et al., 2019] identify the main function of SDNs as decoupling the data plane and control plane. They summarize prior research on the controller placement problem into four master categories: latency-oriented, reliability-oriented, cost-based, and multi-objective. They also identify controller-placement as one of the hottest topics in SDN. [Killi et al., 2018] propose a network partition using a controller placement algorithm based on a mixture of k-means and game theoretic initializations. [Liao et al., 2017] propose a density based controller placement which uses a clustering algorithm to split the network into multiple sub-networks. [Papa et al., 2018] consider ECP in the context of satellite networks and study the use-case scenario of SDN-enabled satellite space segments. They design an integer linear program to address this problem. Focusing on reliability aspects of ECP, [Alshamrani et al., 2018] address maximizing fault-tolerance aspects of controller placement rather than performance. They show sacrificing latency for reliability is generally not a good trade-off except in special cases.

[Das and Gurusamy, 2018] use a multi-objective optimization model to derive a multi-period roll-out plan for controller placements. A similar problem to ECP, the satellite gateway placement problem, is addressed in detail in [Liu et al., 2018]. [Zhiyang Su and Hamdi, 2015] propose a novel scheme to minimize measurement overhead, and formulate the Measurement-aware Distributed Controller Placement (MDCP) problem as a quadratic integer programming problem.

[Jalili et al., 2019] consider an Analytic Hierarchy Process (AHP) to address the multi-criteria controller assignment problem. Apart from latency they also address hop count and link utilization as part of the controller assignment process and use a hybridized ad-hoc genetic algorithm to solve it. [Zhang et al., 2018] design a multi-objective controller placement scheme that simultaneously addresses reliability, load balance and low latency. They use the heuristic adaptive bacterial foraging optimization to solve this problem. [Tao et al., 2018] derive the specific position of all network controllers by minimizing a linear function of load balance factor and total flow request cost. [Dvir et al., 2018] study

the wireless controller placement problem using a multi-objective optimization problem and measure the sensitivity of this placement to variant metrics.

In this paper we present the first maximum entropy based clustering algorithm to address ECP in wireless edge networks. A tutorial on deterministic annealing for the unfamiliar reader may be found in [Rose, 1998]. We distinguish our algorithms from previous clustering approaches in that it is the first multi-objective clustering approach to the ECP problem and it does not require initialization. We found previous algorithms in literature that typically enjoy a fast speed such as Cuckoo search, GA, and other ad-hoc heuristics suffering from susceptibility to poor local optima solutions. On the other hand exact approaches like quadratic integer programming are too slow to be practical for real-case scenarios. Our algorithms address these shortcomings by leveraging their ability to sense and escape poor local minima and at the same time enjoy fast speed due to linear computational complexity in terms of parameters of the problem.

III. PROBLEM STATEMENT

Wireless networks can be illustrated by a graph as shown in Figure 2, in which the vertices are the network nodes and the edges represent the communication between them. One or multiple numbers of these vertices can be designated as a controller, where the optimal number and placement of these controllers depends on how far and how close graph vertices are from each other in terms of communication delay associated with graph edges. ECP reduces to finding this optimal assignment of controllers. In this scheme both the nodes and controllers they are assigned to and the controllers themselves constantly communicate data. This means that a scattered placement of controllers may reduce the delay cost but increase the synchronization cost. On the contrary a more compact placement of edge controllers can reduce the synchronization cost while increasing the delay cost between nodes and the controllers. Two dominant schemes for placement of controllers typically considered are leader-less and leader-based [Qin et al., 2018]. The distinction between the two is that in the former all pairs of controllers in the network directly communicate with each other while in the latter controllers only communicate with a leader controller.

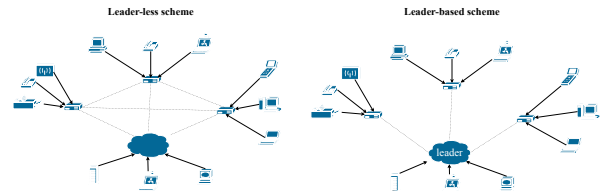


Fig. 2. Leader-based versus Leader-less edge controller placement scheme

To cast this problem as a mathematical program we define \mathcal{N} as the set of all edge nodes with $\text{Card}(\mathcal{N}) = N$ and \mathcal{N}_h as the set of edge nodes that can serve as controllers with

$\mathcal{N}_h \subseteq \mathcal{N}$. Additionally, $\mathcal{X} = (x_i \in \mathbb{R}^d, i \in \mathcal{N})$ determines the position of edge nodes in the wireless network. We use $\bar{\mathcal{X}} = (\bar{x}_i \in \{0, 1\}, i \in \mathcal{N}_h)$ to represent the controller placement policy. If we choose node i to play the role of a controller then $\bar{x}_i = 1$ otherwise $\bar{x}_i = 0$. Similarly $\mathcal{Q} = (q_{ij} \in \{0, 1\}, i \in \mathcal{N}, j \in \mathcal{N}_h)$ determines the controller assignment policy where $q_{ij} = 1$ if node i is assigned to controller j otherwise $q_{ij} = 0$. $\mathcal{Z} = (z_j \in \{0, 1\}, j \in \mathcal{N}_h)$ determines the leader assignment policy in the leader-based scheme. $z_j = 1$ if controller j is the leader and $z_j = 0$ otherwise. $d_{ij} = d(x_i, x_j)$ encodes the communication delay between nodes i and j which we assume to be proportional to the squared Euclidean distance, i.e. $d_{ij} = \|x_i - x_j\|_2^2$.

A. Leader-less Case

In this setting all controllers communicate not only with edge nodes but also with each other. Thus we incur a controller synchronization cost between *all* pairs of controllers. We can express the optimal assignment as the solution of the following integer program:

$$\min_{\mathcal{Q}, \bar{\mathcal{X}}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_h} q_{ij} d_{ij} + \gamma \sum_{i, j \in \mathcal{N}_h} \bar{x}_i \bar{x}_j d_{ij} \sum_{k \in \mathcal{N}} q_{kj} \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_h} q_{ij} = 1 \quad \forall i \in \mathcal{N} \quad (2)$$

$$q_{ij} \leq \bar{x}_j \quad \forall i, j \in \mathcal{N} \quad (3)$$

$$\bar{x}_i \in \{0, 1\}, \quad i \in \mathcal{N}_h \quad (4)$$

$$q_{ij} \in \{0, 1\}, \quad i \in \mathcal{N}, j \in \mathcal{N}_h, \quad (5)$$

The first term in the objective function corresponds to communication delay across all node-controller pairs. The second term shows the synchronization delay between controllers. Note that synchronization delay also depends on how many nodes are assigned to a certain controller. Constraint (2) ensures that each edge node is only assigned to one controller; constraint (3) ensures node assignments to a controller are only made to designated controller nodes. Parameter $\gamma \geq 0$ shows the relative importance of controller synchronization delay compared to controller-node delay.

B. Leader-based Case

The leader-based case is similar to the previous one except that controllers synchronize only with the leader. We can express the optimal assignment in this setting as the solution to the following integer program:

$$\min_{(\mathcal{Q}, \bar{\mathcal{X}}, \mathcal{Z})} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_h} q_{ij} d_{ij} + \gamma \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h} \bar{x}_i z_j (N d_{ij}) \quad (6)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}_h} q_{ij} = 1 \quad \forall i \in \mathcal{N} \quad (7)$$

$$q_{ij} \leq \bar{x}_j \quad \forall i, j \in \mathcal{N} \quad (8)$$

$$\sum_{j \in \mathcal{N}_h} z_j = 1 \quad (9)$$

$$\bar{x}_i \in \{0, 1\}, \quad i \in \mathcal{N}_h \quad (10)$$

$$q_{ij} \in \{0, 1\}, \quad i \in \mathcal{N}, j \in \mathcal{N}_h. \quad (11)$$

Constraint (9) ensures that there is always exactly one leader controller in the leader-based setting. Both leader-less and leader-based cases are NP-hard nonlinear combinatorial problems with no guarantees for finding a global optimum solution [Singh and Srivastava, 2018].

IV. SOLUTION APPROACH

We assume that the delay and synchronization costs are all equivalent to the squared Euclidean distance between the network nodes. This is not far-fetched since according to [Qin et al., 2018] these costs are proportional to if not determined by the Euclidean distances. We further assume that geospatial coordinates¹ of the nodes are provided to us instead of the mutual delays between network nodes.

In the deterministic annealing clustering setting, the expected *distortion*² can be defined as

$$D = \sum_{i=1}^N p(x_i) \sum_{j=1}^m p(y_j | x_i) D(x_i, y_j).$$

$X = \{x_i\}_{i=1}^N$ are the data points and $Y = \{y_j\}_{j=1}^m$ are cluster centroids, or edge controller locations, to be determined. $p(y_j | x_i)$ is called the association probability³ of point x_i with centroid y_j and $D(x_i, y_j)$ is the distortion measure which is typically chosen to be the squared Euclidean distance. We interpret $p(x_i)$ as the relative importance given to i th node and assume, if not otherwise indicated that $p(x_i) = \frac{1}{N}$. System entropy can be defined as $H = -\sum_{i=1}^N p(x_i) \sum_{j=1}^m p(y_j | x_i) \log p(y_j | x_i)$. We also define the system free energy as $F = D - TH$ where T is the system's so-called temperature.⁴ Note that F can be viewed as the Lagrangian for the primary objective of minimizing D , with T being the Lagrange multiplier. The central iteration of DA can be summarized as sequentially optimizing F with respect to the free parameters, i.e. association probabilities and centroid locations.

A. Leader-less Case

For the purpose of adapting the DA clustering to the leader-less ECP problem we define the distortion measure as $D(x_i, y_j) = d(x_i, y_j) + \gamma \sum_{j'=1}^m d(y_j, y_{j'})$. This means the distortion between edge node x_i and controller y_j not only depends on the communication delay between these two nodes but also depends on how far the y_j is placed from other controllers $y_{j'}$.

In order to observe the relation to integer program (1)-(5) notice we can write total distortion as

¹Here, assumed to be a two dimensional or three dimensional vector representing the location of each edge node.

²Distortion is an average weighted distance term, between nodes and centroids, that serves as our basic cost function.

³The weighting indicating that a node belongs to a particular centroid. For each node the sum of these associations over all centroids must equal one.

⁴A coefficient scaling the entropy term which indicates how important the entropy term is compared to the distortion term. We typically reduce this coefficient from a high value to a value close to zero.

$$D = \sum_{i=1}^N \sum_{j=1}^m p(y_j | x_i) d(x_i, y_j) \quad (12)$$

$$+ \gamma \sum_{j'=1}^m \sum_{j=1}^m \left(d(y_j, y_{j'}) \sum_{i=1}^N p(y_j | x_i) \right)$$

This is objective function (1) with hard assignments q_{ij} replaced by the soft association probabilities. As described earlier we define the system's free energy as $F = D - TH$. Setting partial derivatives of the free energy term with respect to association probabilities to zero and solving, yields solution:

$$p(y_j | x_i) = \frac{\exp\left(-\frac{D(x_i, y_j)}{T}\right)}{Z_i}, \quad Z_i = \sum_{j=1}^m p(y_j | x_i)$$

Thus association probabilities have the celebrated Boltzmann distribution. Similarly setting derivatives with respect to the centroids y_j to zero leads to the following linear systems of equations:

$$\eta y_j - \gamma \sum_{j' \neq j} y_{j'} = C_j, \quad j = 1, \dots, m \quad (13)$$

where $\eta = \gamma(m-1) + 1$ and $C_j = \sum_{i=1}^N p(x_i | y_j) x_i$. We may compute $p(x_i | y_j)$ using Bayes' rule. This gives us a linear system of md variables and md equations with m and d being respectively the number of centroids and the dimensionality of data. It is essential for the convergence of our clustering algorithm that this linear system of equations always has a solution.

Proposition 1: Given the linear system of equations in (13) with η and C_j defined as above, if $\gamma \neq \frac{1}{n-m}, \frac{1}{n-2m}$ then there always exists a unique solution $\{y_j\}_{j=1}^m$, where the coefficient matrix associated with the system of the equations is non-degenerate with determinant $\left(\frac{(\gamma m+1)^m (\gamma(n-m)-1)}{\gamma(n-2m)-1}\right)^d$.

See Appendix for a proof. The resulting DA clustering algorithm for the this case is given in Algorithm 1.

For the convergence test we stop at iteration τ if $\|F_\tau - F_{\tau-1}\| < \delta$ for some predetermined tolerance level δ . In the last line of Algorithm 1 we designate the closest valid node to each centroid as a controller.

The iteration complexity for this algorithm depends on (a) calculation of mutual squared Euclidean distances between x_i, y_j for $i \in \{1, \dots, N\}, j \in \{1, \dots, m\}$, (b) similar calculation of mutual distances between centroids, (c) calculation of association probabilities and (d) solving the linear system of equations. The complexities for these operations are respectively, $O(NK_{max}d)$, $O(K_{max}^2d)$, $O(K_{max}N)$ and $O(K_{max}^3d^3)$. For large N these terms are dominated by $O(NK_{max}d)$, thus for a maximum number of iterations τ the algorithmic computational complexity for the leader-less case is $O(\tau NK_{max}d)$ which is linear in data size, maximum number of clusters and dimensionality of data.

Algorithm 1: ECP-LL

Set max # of clusters K_{max} and min temperature T_{min} ;

Initialize: $T \rightarrow \inf, K = 1, y_1 = \sum_{i=1}^N x_i p(x_i)$;

while Convergence test **do**

Update:

$$p(y_j | x_i) \leftarrow \exp\left(-\frac{d(x_i, y_j) + \gamma \sum_{j'=1}^m d(y_j, y_{j'})}{T}\right) / Z_i$$

Solve:

$$\eta y_j^{new} - \gamma \sum_{j' \neq j} y_{j'}^{new} = \sum_{i=1}^N p(x_i | y_j) x_i, \quad j = 1, \dots, m$$

Update: $y_j \leftarrow y_j^{new} \quad j = 1, \dots, m$;

if $T \leq T_{min}$ **then**

break;

else

Cooling Step: $T \leftarrow \alpha T (\alpha < 1)$;

Generate small random vector ϵ

Replace y_j with $y_j + \epsilon$ and $y_j - \epsilon$;

end

end

Perform last step iteration for $T = 0$;

$y_j \leftarrow \arg \min_{x_i \in \mathcal{N}_h} d(x_i, y_j)$;

B. Leader-based case

In order to adapt DA to the leader-based ECP problem we define an appropriate distortion measure by:

$$D(x_i, y_j) = d(x_i, y_j) + \gamma \min_{j \in \{1, \dots, m\}} \sum_{j'=1}^m d(y_j, y_{j'}) \quad (14)$$

Similarly we can consider the weighted total distortion as:

$$D = \sum_{i=1}^N \sum_{j=1}^m p(y_j | x_i) d(x_i, y_j) + \gamma \min_{j \in \{1, \dots, m\}} \sum_{j'=1}^m N d(y_j, y_{j'}) \quad (15)$$

In order to observe its relation to MINLP objective function, notice (6) is equivalent to the following objective function:

$$\min_{(\mathcal{Q}, \bar{X})} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_h} q_{ij} d_{ij} + \gamma \min_{j \in \{1, \dots, m\}} \sum_{i \in \mathcal{N}_h} \bar{x}_i (N d_{ij})$$

To establish this equivalence, we used the relationship that for $W = \{w_i\}_{i=1}^m$ and $S = \{W \in \mathbb{R}_+^n \mid \sum_{i=1}^m w_i = 1\}$ then $\min_{Z \in S} \sum_{j=1}^m z_j \alpha_j = \min_{j \in \{1, \dots, m\}} \alpha_j$.

We define the system's free energy similarly to the previous case. Setting the gradient with respect to the association probabilities to zero, yields solution:

$$p(y_j | x_i) = \frac{\exp\left(-\frac{d(x_i, y_j)}{T}\right)}{Z_i}, \quad Z_i = \sum_{j=1}^m p(y_j | x_i)$$

Denote $j^* = \arg \min_{j \in \{1, \dots, m\}} \sum_{j'=1}^m d(y_j, y_{j'})$ as the index of the leader centroid and set gradient with respect to y_j to

zero to yield the centroid update rules:

$$y_j = \frac{\gamma N y_{j^*} + \sum_{i=1}^N p(y_j | x_i) x_i}{\gamma N + \sum_{i=1}^N p(y_j | x_i)} \quad y_j \neq y_{j^*} \quad (16)$$

$$y_{j^*} = \frac{\gamma N \sum_{j' \neq j^*} y_{j'} + \sum_{i=1}^N p(y_{j^*} | x_i) x_i}{(m-1)\gamma N + \sum_{i=1}^N p(y_{j^*} | x_i)} \quad (17)$$

We can compute values of y_j and y_{j^*} by substituting (17) in (16). The resulting DA clustering algorithm for the leader-based case can be found in Algorithm 2.

Algorithm 2: ECP-LB

Set limits: max # of clusters K_{max} and minimum temperature T_{min} ;

Initialize: $T \rightarrow \inf, K = 1, y_1 = \sum_{i=1}^N x_i p(x_i)$;

while Convergence test **do**

Update: $p(y_j | x_i) \leftarrow \exp\left(-\frac{d(x_i, y_j)}{T}\right) / Z_i$;

Solve: $j^* = \arg \min_{j \in \{1, \dots, m\}} \sum_{j'=1}^m d(y_j, y_{j'})$;

Solve:

$$y_j^{new} = \frac{\gamma N y_{j^*} + \sum_{i=1}^N p(y_j | x_i) x_i}{\gamma N + \sum_{i=1}^N p(y_j | x_i)} \quad y_j \neq y_{j^*}$$

$$y_{j^*}^{new} = \frac{\gamma N \sum_{j' \neq j^*} y_{j'} + \sum_{i=1}^N p(y_{j^*} | x_i) x_i}{(m-1)\gamma N + \sum_{i=1}^N p(y_{j^*} | x_i)}$$

Update: $y_j \leftarrow y_j^{new} \quad j = 1, \dots, m$;

if $T \leq T_{min}$ **then**

break;

else

Cooling Step: $T \leftarrow \alpha T (\alpha < 1)$;

Generate small random vector ϵ

Replace y_j with $y_j + \epsilon$ and $y_j - \epsilon$;

end

end

Perform last step iteration for $T = 0$;

$y_j \leftarrow \arg \min_{x_i \in \mathcal{N}_h} d(x_i, y_j)$;

The computational complexity for the leader-based algorithm is similar to the previous one, except for the centroid calculation step in which we no longer have to compute a linear system of equations. The computational complexity is $O(NK_{max}d) + O(K_{max}^2d) + O((N + K_{max})d) + O(NK_{max})$. For a maximum of τ iterations and large N this is again dominated by $O(\tau NK_{max}d)$.

C. Phase Transformation

Our proposed algorithms undergo *phase transition* phenomenon analogous to the DA clustering algorithm [Rose, 1998]. More specifically, the algorithms illustrated in Section III begin with allocating a single centroid $y_j = \sum_{i \in \mathcal{N}} p(x_i) x_i$ at $T = T_{max} (\rightarrow \infty)$. As T is gradually decreased, there is no perceptible change in solution till a critical value of temperature $T = T_{cr1}$ is reached where the number of centroids increases. Again as T decreases further, we observe no perceptible change in the solution, till another critical temperature $T = T_{cr2}$ value is achieved

where the number of centroids once again increases. In fact, the insensitivity of the solution to T between two consecutive critical values allows us to geometrically anneal the temperature in Algorithms 1 and 2 which makes them computationally efficient [Sharma et al., 2012]. We use the second order necessary condition for optimality to determine the explicit values T_{cr} 's where the phase transition occurs. In particular, at $T = T_{max}$, F is a convex function with its global minimum Y satisfying the second order optimality condition $\frac{\partial^2 F^2}{\partial^2 Y} > 0$. As T gradually decreases there occurs an instance $T = T_{cr1}$ where the Hessian $\frac{\partial^2 F^2}{\partial^2 Y}$ loses rank and the number of centroids increases. The following theorem characterizes the temperature T values where phase transitions occur in the case of ECP-LL algorithm.

Theorem 1: The critical value T_{cr} of the temperature for a given set of centroids $\{y_j\}_{j=1}^m$ and corresponding association weights $\{p(y_j | x_i)\}$ in the ECP-LL algorithm is such that

$$\det \left[\sum_{i=1}^N p(x_i) \left(\Lambda_i (1 + m\gamma) + \gamma I - 2\gamma \Gamma_i^T E - 2\frac{1}{T_{cr}} \Theta_i \right) \right] = 0, \quad (18)$$

where Λ_i, Γ_i, E and $\Theta_i \in \mathbb{R}^{md \times md}$ are known in terms of $\{x_i\}_{i=1}^N, \{y_j\}_{j=1}^m$ and $\{p(y_j | x_i)\}$, I is an $md \times md$ identity matrix.

Similarly, the critical temperature values for the ECP-LB algorithm are determinable. See Appendix for a proof.

V. RESULTS

In order to evaluate the performance of these algorithms we compare their final costs with the integer programs (1)-(5) and (6)-(11). We use the state-of-the-art MINLP solver BARON to draw this comparison. We used Gaussian distribution to generate our data with K as the number of Gaussian clusters within the data. During the implementation we perform a grid search over the hyper-parameter space of K_{max} to find its optimum value.

Superior performance of ECP DA-based clustering algorithms can be observed even in small problem instances like in Figure 3 (a) and (b). While BARON is stuck in a poor local optimum with an excessive number of controllers, ECP-LL has managed to achieve a considerably lower objective value with fewer controller placements.

In Figure 4, an immediate result of avoiding controller synchronization cost contributes increasingly to the objective function as γ increases.

Figure 5 shows the trade-off between different hyper-parameters for ECP-LL algorithm. (a) shows as γ increases the optimal objective value also increases and stays relatively constant for very large values of γ . This is due to the fact that for large γ , controller placement becomes more packed and at its extreme we would have only one controller to cancel out synchronization cost. Figure 5 (b) also shows the same pattern that as γ increases ECP-LL places fewer controllers in edge network. Figure 5 (c) shows the optimal value for hyper-parameter K_{max} in ECP-LL algorithm. We validate that the optimal value of K_{max} is the number of inherent clusters in the dataset. Figure 5 (d) Shows the the

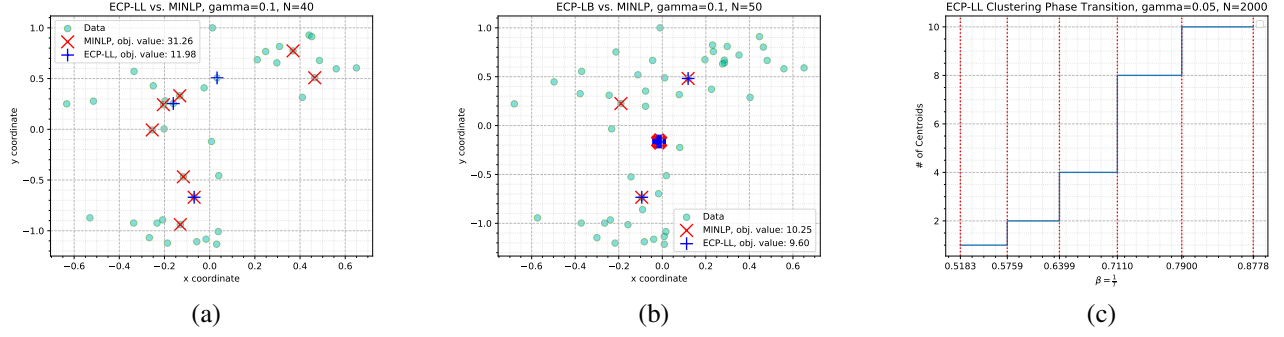


Fig. 3. (a) ECP-LL vs. MINLP (b) ECP-LB vs. MINLP (c) Phase transition phenomenon

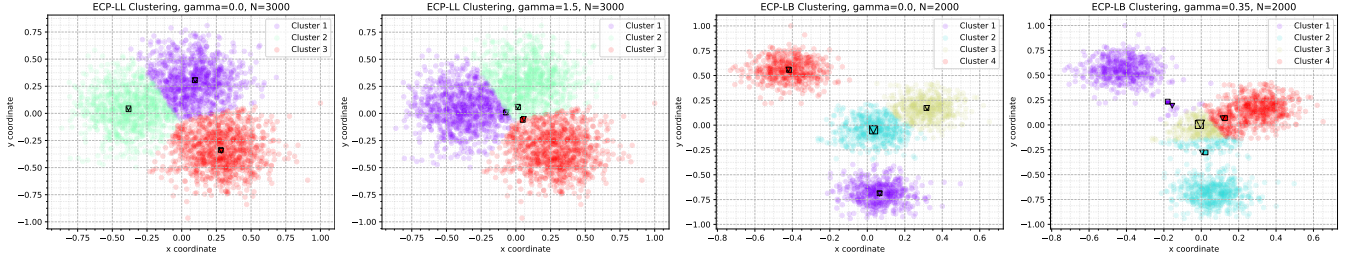


Fig. 4. Controller placement sensitivity to parameter γ

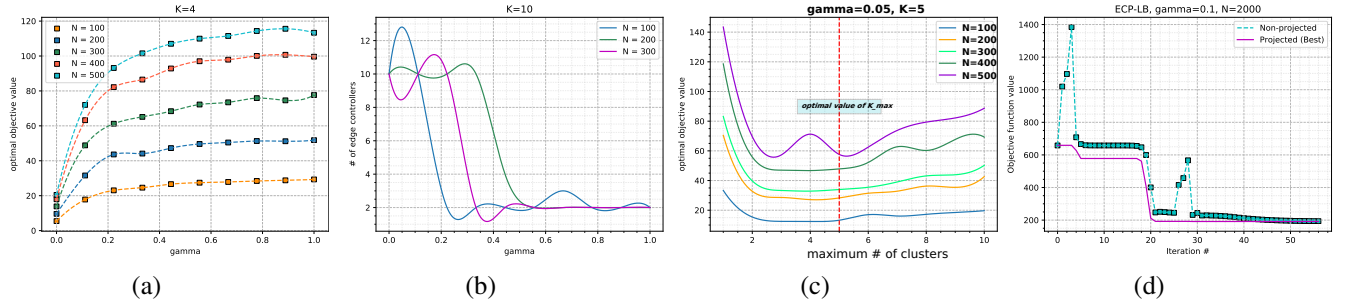


Fig. 5. (a) γ vs. optimal objective value, (b) γ vs. optimal number of controllers, (c) hyper-parameter K_{max} vs. optimal objective value and (d) Iteration number vs. projected and non-projected solutions objective function values

values of non-projected and projected⁵ solutions versus the number of iterations. The projected solution is obtained by setting association probabilities to either zero or one and then projecting the solution centroids onto the data set. We observed that the two converge to the same value across most scenarios. Since the non-projected optimal objective value serves as a lower bound for that of the projected, we can assume ECP-LL has reached, in worst case, a near-optimal solution.

Table I compares performance of ECP-LL against MINLP. While ECP-LL by far outperforms MINLP in terms of total run time, the difference in accuracy is emphasized as problem size increases. ECP-LL and ECP-LB provide consistent performance both in terms of accuracy and speed across different data sizes and varying data clusters as is by design resilient to local minima that riddle the cost function

⁵ At the last step of algorithm we mapped centroids onto the closest edge node available. We call such a solution projected, otherwise we call the solution non-projected.

surface. Figure 6 illustrates how run time grows linearly as a function of data size and number of clusters.

VI. CONCLUSION AND FUTURE WORK

In this work we introduced two multi-objective maximum entropy based clustering algorithms for the problem of Edge Controller Placement in wireless communication networks. ECP-LL and ECP-LB each address a different controller placement topology, and their design is inspired by a Mixed Integer Nonlinear Program. We show that our algorithms outperform state of art MINLP solver, BARON in both speed and accuracy. Total computational complexity for these algorithms is $O(\tau NK_{max}d)$ which is linear in data size, maximum number of clusters and dimensionality of data. As an extension to current work we propose a kernelized version of this algorithm. This is significantly important as in most real-world scenarios only the mutual delays between network nodes are provided (e.g. using latency tests) rather than their geospatial coordinates. Another research avenue is to think

TABLE I

DURATION AND TOTAL COMMUNICATION DELAY AS A FUNCTION OF SIZE OF DATASET N AND NUMBER OF CLUSTERS K WITH $\gamma = 0.1$. TUPLES SHOW COMPLETION TIME (SEC), OBJECTIVE VALUE AND NUMBER OF PLACED CONTROLLERS TRIPLETS. ECP-LL VS BARON

	N=20	N=40	N=60
K=2	(0.31,7.10,2), (622.24,15.37,4)	(0.52,13.24,2), (606.73,37.08,5)	(0.78,17.50,2), (627.56,585.03,55)
K=4	(0.48,8.65,7), (614.54,12.93,5)	(0.77,12.20,7), (610.03,28.41,8)	(1.08,16.82,7), (624.12,233.88,48)
K=6	(0.67,7.75,4), (605.20,13.70,6)	(1.20,15.19,4), (607.32,38.64,9)	(1.61,23.26,4), (638.92,95.92,13)
K=8	(0.87,6.53,4), (1630.94,9.39,6)	(1.34,18.66,4), (606.02,42.28,9)	(1.96,25.82,4), (618.61,390.73,54)
K=10	(1.05,6.81,2), (602.03,10.32,5)	(1.76,12.60,2), (617.59,27.35,10)	(2.49,19.46,4), (621.64,81.89,15)

TABLE II

DURATION AND TOTAL COMMUNICATION DELAY AS A FUNCTION OF SIZE OF DATASET N AND NUMBER OF CLUSTERS K AND $\gamma = 0.1$. TUPLES SHOW COMPLETION TIME (SEC), OBJECTIVE VALUE AND NUMBER OF PLACED CONTROLLERS TRIPLETS. ECP-LB VS BARON

	N=20	N=40	N=60
K=2	(0.93,1.99,5), (2.11,6.21,4)	(1.63,3.68,5), (373.22,11.88,4)	(2.36,6.22,2), (906.84,15.24,3)
K=4	(1.49,2.89,6), (14.87,4.34,5)	(2.69,4.58,7), (267.35,6.09,6)	(4.58,6.79,7), (604.13,9.16,6)
K=6	(2.06,3.52,8), (63.59,4.88,3)	(3.71,8.14,5), (305.31,8.85,5)	(5.23,9.10,8), (612.36,12.78,5)
K=8	(2.81,2.71,11), (16.87,3.78,4)	(4.61,8.17,11), (458.29,10.56,5)	(7.06,11.73,11), (604.58,15.07,5)
K=10	(3.40,3.05,8), (68.99,4.90,4)	(7.10,5.04,9), (297.49,7.25,4)	(10.16,7.34,12), (607.64,12.27,5)

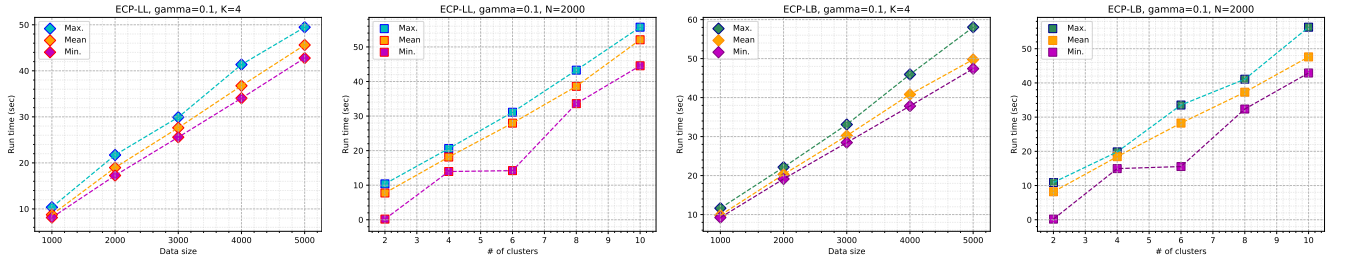


Fig. 6. ECP algorithms run time vs. # of clusters and data size.

of a mechanism to project centroids onto the data set at each iteration, as there is no provable guarantee that all centroids will correspond to a data point. This can potentially yield better solutions upon convergence of the algorithm. Finally one can consider ECP under the assumption that network nodes are mobile, which will require tracking controller design.

APPENDIX

Proof of Proposition 1

We can write the coefficient matrix associated with (13) as the block matrix $\Theta \in \mathbb{R}^{md \times md}$ with diagonal blocks equal to ηI and non-diagonal blocks equal to $-\gamma I$ such that $I \in \mathbb{R}^{d \times d}$. Dividing all rows by constant $-\gamma$ we get $\det(\Theta) = (-\gamma)^{md} \det(\bar{\Theta})$. $\bar{\Theta}$ is a block diagonal matrix with diagonal elements equal to αI and non-diagonal blocks equal to I with $\alpha = -\frac{\eta}{\gamma}$. Using straightforward linear algebra we can

transform $\bar{\Theta}$ to an upper triangular matrix:

$$\bar{\Theta} \times \begin{bmatrix} I & 0 & \dots & 0 \\ \frac{-1}{\alpha+n-2}I & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-1}{\alpha+n-2}I & \frac{-1}{\alpha+n-3}I & \dots & I \end{bmatrix} = \begin{bmatrix} \beta_1 I & \times & \dots & \times \\ 0 & \beta_2 I & \dots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \beta_m I \end{bmatrix} = \Phi \quad (19)$$

Where $\beta_i = \alpha - \frac{n-i}{\alpha+n-i-1}$ and $\det(\bar{\Theta}) = \det(\Phi) = \prod_{i=1}^m (\beta_i)^d$. We can use simple telescoping to further simplify the product to $\left(\frac{(\alpha-1)^m (\alpha+n-1)}{\alpha+n-(m+1)} \right)^d$. This will give $\det \Theta = \left(\frac{(\gamma m+1)^m (\gamma(n-m)-1)}{\gamma(n-2m)-1} \right)^d$ which is well defined for $\gamma \neq \frac{1}{n-m}, \frac{1}{n-2m}$.

Proof of Theorem 1

We use the calculus of variations to determine the second order optimality condition for F . In particular, consider a given set of optimal centroids $Y = [y_1^T, \dots, y_m^T]^T \in \mathbb{R}^{md}$ and $Y + \epsilon \Psi$ to be the corresponding set of perturbed centroids where $\Psi = [\psi_1^T, \dots, \psi_m^T]^T \in \mathbb{R}^{md}$ is the perturbation vector. The second order condition for optimality states that $\left. \frac{\partial^2 F^2(Y + \epsilon \Psi)}{\partial Y^2} \right|_{\epsilon=0} > 0$ for all possible perturbations Ψ . We

obtain:

$$\frac{\partial F^2(Y + \epsilon\Psi)}{\partial Y^2} = 2\Psi^T \sum_{i=1}^N p(x_i) \left[\Lambda_i(1 + m\gamma) + \gamma I \right. \\ \left. - 2\gamma\Gamma_i^T E - 2\frac{1}{T}\Theta_i \right] \Psi + 4\beta\Psi^T \left(\sum_{i=1}^N p_i \mathcal{L}_i^T \mathcal{L}_i \right) \Psi, \quad (20)$$

where $\Lambda_i = \text{diag}(p(y_1|x_i) \ p(y_2|x_i) \ \dots \ p(y_m|x_i))$, m is the number of centroids, $E = 1_m^T \otimes I_d$, $1_m \in \mathbb{R}^m$ is a vector of 1's, $\Gamma_i = \sum_{j=1}^m p(y_j|x_i) E_j \Psi$, $E_j = e_j^T \otimes I_d$, e_j is the basis vector in \mathbb{R}^m with j -th entry as 1, $\Theta_i = \sum_{j=1}^m p(y_j|x_i) T_{ji}^T T_{ji}$, $T_{ji} = (y_j - x_i)^T E_j + \gamma \sum_{k=1}^m (y_j - y_k)^T (E_j - E_k)$, and $\mathcal{L}_i = \sum_{j=1}^m p(y_j|x_i) T_{ji}$. We claim that the Hessian in (20) is positive for all perturbations Ψ if and only if the first term in (20) is positive. Clearly, when the first term is positive this is true owing to the non-negativity of the second term $((\mathcal{L}_i \Psi)^T (\mathcal{L}_i \Psi) \geq 0)$. This establishes the *if* part of our claim. For the *only if* part we show that there exists a non trivial perturbation when the matrix in the first term is not positive definite for which the second term becomes zero. In fact, let y_{j_0} be the centroid that splits into two further centroids. Let $\psi_k = 0 \ \forall \ k \neq j_0$ and ψ_{j_0} such that $p(y_{j_0}|x_i)(y_{j_0} - x_i)^T \psi_{j_0} = 0$ (a non trivial ψ_{j_0} exists); this choice of perturbation results in the second term being zero. Thus, whenever the matrix in the first term loses rank we can construct a perturbation that makes the second term zero. This is the phase transition condition in (18).

REFERENCES

- [Alshamrani et al., 2018] Alshamrani, A., Guha, S., Pisharody, S., Chowdhary, A., and Huang, D. (2018). Fault Tolerant Controller Placement in Distributed SDN Environments. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE.
- [Das and Gurusamy, 2018] Das, T. and Gurusamy, M. (2018). INCEPT: INcremental ControllEr PlacemEnT in Software Defined Networks. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE.
- [Dungay, 2016] Dungay, D. (2016). Software Defined Networking (SDN) Explained — Comms Business.
- [Dvir et al., 2018] Dvir, A., Haddad, Y., and Zilberman, A. (2018). Wireless controller placement problem. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE.
- [Heller et al., 2012] Heller, B., Sherwood, R., and McKeown, N. (2012). The controller placement problem. In *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*, page 7, New York, New York, USA. ACM Press.
- [Jalili et al., 2019] Jalili, A., Keshtgari, M., Akbari, R., and Javidan, R. (2019). Multi criteria analysis of Controller Placement Problem in Software Defined Networks. *Computer Communications*, 133:115–128.
- [Killi et al., 2018] Killi, B. P. R., Reddy, E. A., and Rao, S. V. (2018). Cooperative game theory based network partitioning for controller placement in SDN. In *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, pages 105–112. IEEE.
- [Kuang et al., 2018] Kuang, H., Qiu, Y., Li, R., and Liu, X. (2018). A Hierarchical K-Means Algorithm for Controller Placement in SDN-Based WAN Architecture. In *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 263–267. IEEE.
- [Li and Xu, 2018] Li, F. and Xu, X. (2018). A Discrete Cuckoo Search Algorithm for the Controller Placement Problem in Software Defined Networks. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 292–296. IEEE.
- [Liao et al., 2017] Liao, J., Sun, H., Wang, J., Qi, Q., Li, K., and Li, T. (2017). Density cluster based approach for controller placement problem in large-scale software defined networkings. *Computer Networks*, 112:24–35.
- [Liu et al., 2018] Liu, J., Shi, Y., Zhao, L., Cao, Y., Sun, W., and Kato, N. (2018). Joint Placement of Controllers and Gateways in SDN-Enabled 5G-Satellite Integrated Network. *IEEE Journal on Selected Areas in Communications*, 36(2):221–232.
- [Lu et al., 2019] Lu, J., Zhang, Z., Hu, T., Yi, P., and Lan, J. (2019). A Survey of Controller Placement Problem in Software-Defined Networking. *IEEE Access*, 7:24290–24307.
- [Papa et al., 2018] Papa, A., De Cola, T., Vizarreta, P., He, M., Mas Machuca, C., and Kellerer, W. (2018). Dynamic SDN Controller Placement in a LEO Constellation Satellite Network. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 206–212. IEEE.
- [Qin et al., 2018] Qin, Q., Poularakis, K., Iosifidis, G., Kompella, S., and Tassiulas, L. (2018). SDN Controller Placement With Delay-Overhead Balancing in Wireless Edge Networks. *IEEE Transactions on Network and Service Management*, 15(4):1446–1459.
- [Rose, 1998] Rose, K. (1998). Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems. Technical report, California Institute of Technology.
- [Sharma et al., 2012] Sharma, P., Salapaka, S. M., and Beck, C. L. (2012). Entropy-based framework for dynamic coverage and clustering problems. *IEEE Transactions on Automatic Control*, 57(1):135–150.
- [Singh and Srivastava, 2018] Singh, A. K. and Srivastava, S. (2018). A survey and classification of controller placement problem in SDN. *International Journal of Network Management*, 28(3):e2018.
- [Tao et al., 2018] Tao, P., Ying, C., Sun, Z., Tan, S., Wang, P., and Sun, Z. (2018). The Controller Placement of Software-Defined Networks Based on Minimum Delay and Load Balancing. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 310–313. IEEE.
- [Zhang et al., 2018] Zhang, B., Wang, X., and Huang, M. (2018). Multi-objective optimization controller placement problem in internet-oriented software defined network. *Computer Communications*, 123:24–35.
- [Zhiyang Su and Hamdi, 2015] Zhiyang Su and Hamdi, M. (2015). MDCP: Measurement-Aware Distributed Controller Placement for Software Defined Networks. In *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, pages 380–387. IEEE.