frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 01 November 2016
doi: 10.3389/fams.2016.00020

Machine Learning Meliorates Computing and Robustness in Discrete Combinatorial Optimization Problems

Fushing Hsieh [1]*, *Kevin Fujii* [1] *and Cho-Jui Hsieh* [2]

[1] *Department of Statistics, University of California, Davis, Davis, CA, USA,* [2] *Departments of Computer Science and Statistics, University of California, Davis, Davis, CA, USA*

Discrete combinatorial optimization problems in the real world are typically defined via an ensemble of potentially high dimensional measurements pertaining to all subjects of a system under study. We point out that such a data ensemble in fact embeds with system's information content that is not directly used in defining the combinatorial optimization problems. Can machine learning algorithms extract such information content and make combinatorial optimizing tasks more efficient? Would such algorithmic computations bring new perspectives into this classic topic of Applied Mathematics and Theoretical Computer Science? We show that answers to both questions are positive. One key reason is due to permutation invariance. That is, the data ensemble of subjects' measurement vectors is permutation invariant when it is represented through a subject-vs.-measurement matrix. An unsupervised machine learning algorithm, called Data Mechanics (DM), is applied to find optimal permutations on row and column axes such that the permuted matrix reveals coupled deterministic and stochastic structures as the system's information content. The deterministic structures are shown to facilitate geometry-based divide-and-conquer scheme that helps optimizing task, while stochastic structures are used to generate an ensemble of mimicries retaining the deterministic structures, and then reveal the robustness pertaining to the original version of optimal solution. Two simulated systems, Assignment problem and Traveling Salesman problem, are considered. Beyond demonstrating computational advantages and intrinsic robustness in the two systems, we propose brand new robust optimal solutions. We believe such robust versions of optimal solutions are potentially more realistic and practical in real world settings.

Keywords: data mechanics, assignment problem, traveling salesman problem, robustness, network mimicking

OPEN ACCESS

Edited by:
Yiming Ying,
University at Albany, SUNY, USA

Reviewed by:
Junhong Lin,
Massachusetts Institute of Technology and Istituto Italiano di Tecnologia, Italy
Zheng-Chu Guo,
Zhejiang University, China

***Correspondence:**
Fushing Hsieh
fhsieh@ucdavis.edu

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *07 September 2016*
Accepted: *13 October 2016*
Published: *01 November 2016*

Citation:
Hsieh F, Fujii K and Hsieh C-J (2016) Machine Learning Meliorates Computing and Robustness in Discrete Combinatorial Optimization Problems. Front. Appl. Math. Stat. 2:20. doi: 10.3389/fams.2016.00020

1. INTRODUCTION

Discrete combinatorial optimization is a topic in Applied Mathematics and Computer Science. This topic consists of finding an optimal combination upon all involved subjects within a study system. When a real world system is under study, each of its subject is characterized by a vector of measurements. This ensemble of measurement vectors is used to define a discrete combinatorial optimization problem of interest. Among many of these problems, two well-known discrete combinatorial optimization problems are: Assignment problem (AP) and Traveling Salesman problem (TSP). A generic Assignment Problem [1–3] is defined on a $n \times n$ bipartite matrix representing a system consisting of n subjects, each of which is characterized by a n-dim vector of cost of performing n different tasks. While Traveling Salesman problem is defined generically on a symmetric distance matrix between all "cities." An optimal salesman's route, a closed single loop of a group of cities in a form of a polygon, is sought to minimize its perimeter [4–8].

A data matrix, on which combinatorial optimization problems are defined, is taken to be deterministic, as is its optimal solution. It is worth pointing out that this deterministic viewpoint of the data matrix is an extreme concept well taken in classic Applied Mathematics and Computer Science literature. Since greedy exhaustive search schemes are usually not feasible when the size of system is big, many algorithms have been precisely developed and thoroughly studied in terms of mathematical efficiency and computational complexity. Recently in Random Matrix theory of Mathematics and theoretical statistical physics [9–11], a data matrix is a realization of a random matrix. In other words, the idea of data matrix is simply given up for a pure array of random variables. Such a random matrix could be either an unitary matrix [9], the product with its own transpose matrix being equal to the identity matrix, or simply consisting of completely independent random variables. Likewise Random Assignment problem [12–14] and randomly generated cities for TSP [15, 16], or random graph theory [17, 18] are considered and studied.

The random matrix theory now has become an important topic in mathematics, statistics and physics. In the new kind of statistical mechanics based on the theory, the realization of the system is not relevant [19]. That is, the concept of ensemble of system's microstates is no longer essential, since the distributional mechanism being responsible for generating the data matrix is assumed known. Any specific features observed on a data matrix becomes stochastic noise, and what is essential is the averaging patterns resulting from the distributional assumptions. Thus, such a complete random concept of a data matrix is regarded as another extreme viewpoint. Most often it is adopted primarily for theoretic developments, rather than for practical and realistic utilities.

In real world systems, measurements contained in data matrices are likely prone to intrinsic variations, or even errors. Hence it is advantageous to take an observed data matrix as a system's microstate at the time point of study. In the Assignment problem, a microstate can be represented by arranging subjects and tasks along the row and column axes, respectively. Certainly such a microstate is invariant with respect to row and column permutations. When similar subjects and similar tasks are grouped together, then there would be block patterns revealed upon the matrix lattice. It is clear that such block patterns have nothing to do with the definition of Assignment problem, neither with its algorithmic optimal solution. But don't such block patterns evidently depict certain global information about the system of concern?

Likewise in TSP, a symmetric distance matrix represents a microstate of a system under study. Supposed that all involving cities are embedded upon a geographic geometry, that is, some cities are closer to some other cities, but farther away from others. When a permutation embeds with such geometric information, again it would help reveal block patterns within the distance matrix. Such block patterns would not change the Traveling Salesman Problem, nor would it affect its optimal solution. Nonetheless shouldn't such system's global information provide extra and different aspects regarding to TSP?

In this paper we consider and discuss discrete combinatorial optimization problems from a perspective of system information content. Specifically we attempt to address the following question: Can machine learning algorithms extract system's global information content and make combinatorial optimizing tasks more efficient? Here "system information content" refers to patterns pertaining to system's lowest energy macroscopic state, which is in contrast with the system's ensemble of microscopic states. The principle of statistical physics confirms that any microscopic state has to conform to the system's macroscopic state. Though this macrostate's implicit nature prevents it from playing any role in defining a specific discrete combinatorial optimization problem, nonetheless many of its information patterns are indeed computable via machine learning algorithms. We demonstrate that the system's information content is relevant and critical.

The machine learning paradigm employed here to build the block patterns on a matrix lattice is called Data Mechanics (DM), developed in a series of papers [20–23]. Such block patterns are indeed multiscale in the sense of being framed by two Ultrametric trees on row and column axes, respectively. Such multiscale block patterns constitute the deterministic structures, while uniform randomness within each core block jointly constitutes the stochastic structures. The coupling of these two kinds of structures are termed coupling geometry. This concept of a data matrix is right in the middle of the two extreme viewpoints: being complete deterministic, on one hand, and being complete random, on the other hand. In fact such a concept is coherent with information content of a physical system [24], and at the same time is a 2D extension of Kolmogorov complexity [25, 26]. The chief merit of such a coupling geometry is its capability of mimicking the observed data matrix, while retaining the same deterministic and stochastic structures on each copy of mimicry. Hence a mimicry is a microstate, while the coupling geometry prescribes the system's macrostate. An ensemble of microstates would allow us to evaluate practicality and robustness of an optimal solution. In fact these two properties become visible through the block patterns. Further such an ensemble makes possible to address a natural question: Can a non-robust optimal solution be modified and improved?

Further it is intuitive and obvious that algorithmic computing can be made more efficient by adopting the block patterns in these two systems. The theme underpinning computing algorithms developed here is "divide-and-conquer." Although this type of data-driven approach is new in combinatorial optimization, a similar idea has been used recently in machine learning for solving continuous optimization problems. In those problems, the objective function can be written as a summation of loss defined on each training sample, and a family of divide-and-conquer algorithms have been proposed for speeding up the optimization procedure. The main idea of these algorithms is to divide the problem into smaller subproblems by clustering the data, so that each subproblem can be solved independently and efficiently. Among them, Hsieh et al. [27] discussed the divide-and-conquer for kernel Support Vector Machines (SVM), Hsieh et al. [28] applied it to graphical model estimation, Si et al. [29] applied a community detection algorithm to speed up singular value decomposition of sparse graphs, and Mackey et al. [30] used the algorithm for matrix decomposition. Although our paper is totally different from these works, it is interesting to see that the data-driven paradigm is not only important for discrete optimization but also useful in continuous optimization, and we expect the similar idea can be used in many other problems.

At the end of this section we make a final remark regarding algorithmic computations on mimicking. At this stage there are algorithms available in literature that can effectively carry out block-wise uniformity by subject to sequences of row and column sums (or degrees) for binary matrices [31]. For weighted matrices, generative algorithms, such as those proposed in Chen et al. [32] and Barvinok [33], achieve constraints of row and column sums sequences. We must note that these algorithms critically fail to satisfy algorithmic sufficiency criteria. which refer to constraints of two sequences of row and column empirical distributions. An algorithm designed to achieve this criteria is recently developed and published in Fushing et al. [23]. This newly developed algorithm would be given after laying out principle ideas of DM for completeness and convenience purpose.

2. DATA MECHANICS

Let an observed $m \times n$ matrix be denoted as $\mathcal{M}_0^a$ with superscript $a \in \{AP, TSP\}$. The two data matrices are square ones, i.e., $m = n$. The one for AP is asymmetric, so it is a bipartite network. The one for TSP is symmetric, so it is an undirected network. Further let $\mathcal{U}_m$ and $\mathcal{V}_n$ be the permutation groups on the row and column axes, respectively. A permutation on row axis in $\mathcal{U}_m$ is denoted as $\sigma = (\sigma(1), \sigma(2), ...\sigma(m))$, while a member of $\mathcal{V}_n$ is denoted as $\pi = (\pi(1), \pi(2), ...\pi(m))$. Data Mechanics for computing a coupling geometry on $\mathcal{M}_0^a$ consists of a series of computational algorithms. For detailed technical developments, readers are referred to original references [22, 23]. Here we give a briefly review starting from its core device, called Data Cloud Geometry [20, 21].

2.1. Data Cloud Geometry (DCG) Algorithm

1 [Goal:] The goal of DCG algorithmic computing is to build a Ultrametric tree based on a distance matrix, which is either a data matrix by itself as in TSP, or derived from pairwise row or column vectors of $\mathcal{M}_0^a$ via an empirical measure. Denote a generic distance matrix as $\mathcal{D} = [d_{ij}]$.

2 [Temperature regulated similarity matrix:] Covert $\mathcal{D}^a$ into a temperature regulated similarity matrix via heat kernel $\mathcal{S}[T] = [e^{-d_{ij}/T}]$. A small T would enlarge the differences among $\{d_{ij}\}$, while large T would ignore that. Here T is a scaling parameter for aggregating nodes into core clusters or conglomerate ones. This scaling device functions like the resolution tuning in microscope for viewing distinct structures under different resolutions.

3 [Regulated Markovian random walks:] A similarity matrix $\mathcal{S}[T]$ is further converted into a transition probability matrix to govern a Markov random walk within a node-space. In order to effectively and fully explore the whole node-space, our regulated Markov random walk is specially designed by removing a node from the node-space whenever it has accumulated visits up to a pre-selected threshold. As such a random walk going along the discrete time axis, less and less nodes remain in its exploration domain. Therefore, our regulated random walks will not be trapped in a locality of the node-space.

4 [Node-removal time series and its profile:] From the start of a regulated random walk, the recurrent time of node removal is recorded. By plotting this recurrent time series with respect to the chronic order of moved nodes, a profile of spikes is typically observed. A spike indicates the fact that such a random walk enters a new cluster formed with respect to T. That is, an exploration of a regulated random walk indeed extracts the most important geometric information: **nodes removed between two spikes are in the same cluster**. Thus, an exploration of a regulated random walk would give rise to a symmetric binary relational matrix of which node is sharing with which in a cluster.

5 [Ensemble matrix:] By repeatedly performing such data-driven explorations via the regulated random walk on a node-space for many times, we pool all binary relational matrices into an ensemble matrix $\mathcal{E}[T]$, which then becomes the cluster-sharing probability matrix with respect to the temperature T for all involving nodes in the node-space.

6 [Determining key temperatures:] The visible characteristic feature of $\mathcal{E}[T]$ is a series of blocks located along the diagonal. Supposedly each block should be identified as a cluster, and accordingly corresponds to a non-zero eigenvalue of $\mathcal{E}[T]$. Thus, we plot the eigenvalues of $\mathcal{E}[T]$ from largest to smallest to check whether a temperature T gives rise to a clear pattern of non-zero vs. zero. If yes, then we decide how many clusters are realized. By performing this key temperature selection manually over a range of potential temperatures. Each key temperature is selected with a distinct number of clusters. So a series of decreasing numbers is resulted.

7 [Synthesizing an Ultrametric tree:] For a key temperature T and a decided number of clusters, we can recover the cluster memberships by applying one of many available clustering methodologies, such as spectral clustering algorithm and others, on $\mathcal{E}[T]$. We then synthesize the serial configurations of cluster memberships into an Ultrametric tree. Denote such a tree as $\mathcal{T}_{ree}^a$.

The DCG algorithmic computations of Ultrametric tree $\mathcal{T}_{ree}^a$ demonstrates the fact that there exists multiscale structural information contained within a distance matrix $\mathcal{D} = [d_{ij}]$. In contrast, patterns resulted from popular principle component analysis (PCA), factor analysis and Multidimensional scaling

(MDS) method are likely not well tuned with respect to any key temperature, so potentially "out of focus," on one hand. On the other hand, the popular hierarchical clustering (HC) algorithm based on the same distance matrix $\mathcal{D} = [d_{ij}]$ would produce a (HC)tree having too many levels.

For a rectangular data matrix, or a bipartite network, it involves two node-spaces $\mathcal{X}$ and $\mathcal{Y}$, which are respectively arranged on column and row axes. Hence two DCG Ultrametric trees can be built upon each of the two axes. These Ultrametric trees are to be tightly coupled together in order to successfully reveal the interacting relational patterns between $\mathcal{X}$ and $\mathcal{Y}$ embedded with in the data matrix $\mathcal{M}^a_0$. The Data Mechanics algorithm is developed to achieve this goal.

2.2. Data Mechanics

1 [Goal:] To build two coupled DCG Ultrametric trees in order to achieve the discovery of interacting relational patterns between $\mathcal{X}$ and $\mathcal{Y}$.

2 [Iterative computing-I:] Begining with an empirical distance measure, we derive a distance matrix on a node-space, say $\mathcal{Y}$. Then we compute an initial DCG Ultrametric tree $\mathcal{T}^{a(0)}_{ree}(\mathcal{Y})$. Next we construct an distance measure on node-space $\mathcal{X}$ by adapting in the tree $\mathcal{T}^{a(0)}_{ree}(\mathcal{Y})$. Specially this distance should accommodate discrepancies from component-wise as well as multiple cluster-wise aspects. That is, this adaptive distance measure is constructed in such a fashion that several clustering compositions on different tree levels are taken into consideration. Then a DCG Ultrametric tree $\mathcal{T}^{a(1)}_{ree}(\mathcal{X})$ on node-space $\mathcal{X}$ is built.

3 [Iterative computing-II:]Then we update the initial distance measure on $\mathcal{Y}$ with respect to tree $\mathcal{T}^{a(1)}_{ree}(\mathcal{X})$, and then build a revised DCG Ultrametric tree $\mathcal{T}^{a(1)}_{ree}(\mathcal{Y})$. Repeat such an iterative computing scheme until both DCG trees are stabilized. Denote both trees as $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$, respectively.

4 Let permutation $\pi^* \in \mathcal{V}_n$ be conforming to $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and permutation $\sigma^* \in \mathcal{U}_m$ to $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$. The permuted matrix $\sigma^*\mathcal{M}^a_0\pi^*$ would reveal multiscale block patterns due to similar rows and columns are iteratively grouped together. In fact this permutation pair (σ^*, π^*) is close to the optimal one which achieves the minimum total variation on the $m \times n$ matrix lattice with a 4-node-neighborhood system.

5 [Deterministic structures:] The block patterns embedded in $\sigma^*\mathcal{M}^a_0\pi^*$, which are jointly framed by $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$, is taken as the deterministic structures embedded within the original bipartite network data represented by matrix $\mathcal{M}^a_0$.

The distance adaptation is the key step for building the coupling relationships between the two DCG Ultrametric trees $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$. This is the reason why the interacting relational patterns between node-spaces $\mathcal{X}$ and $\mathcal{Y}$ can be revealed as multiscale block patterns through the permuted matrix $\sigma^*\mathcal{M}^a_0\pi^*$. In contrast, such pattern information would be missed through applications of Singular value decomposition (SVD) analysis or Support Vector Machines (SVM).

Next consider a mimicking block being framed by one core cluster of $\mathcal{X}$ on the bottom level of $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and one core cluster of $\mathcal{Y}$ on the bottom level of $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$. It is clear that the block-wise uniformity is subject to constraints of its two sequences of row and column empirical distributions on a weighted matrix setting. Such constraints are equivalent to the two sequences of row and column sums in a binary setting [34].

2.3. Bipartite Network Mimicking and Its State Ensemble

1 [Goal:] To discover the inherent randomness within each block framed by two DCG Ultrametric trees $\mathcal{T}^{a(*)}_{ree}(\mathcal{X})$ and $\mathcal{T}^{a(*)}_{ree}(\mathcal{Y})$, and then build the state ensemble, denoted as Ω^a.

2 [Categorizing $\mathcal{M}^a_0$:] Construct a HC tree and an empirical distribution based on the pooled set of entries of $\mathcal{M}^a_0$ and fit an optimal gapped piecewise-linear function onto this empirical distribution function. Make digital categories according to the possibly gapped histogram, and then digitally categorize the entire matrix $\mathcal{M}^a_0$. Denote this matrix of categorical entries as $\hat{\mathcal{M}}^a_0$. The multiscale block patterns in $\sigma^*\hat{\mathcal{M}}^a_0\pi^*$ would become even more evident because the categorizing procedure is also a de-noise procedure.

3 [Mimicking a block via binary slicing procedure:] The uniformity of a block is conceptually captured by constraining with respect to row and column sequences of empirical distributions. To embrace such uniformity, we first slice the entire block into a sequence of binary matrices by thresholding according to the distinct digital categories. Simulate each slice of the binary matrix individually, and then add them all up to form a simulated categorical block.

4 [Mimicking the entire matrix and building Ω^a:] A mimicry of $\sigma^*\mathcal{M}^a_0\pi^*$ is obtained by pitching up all simulated categorical blocks, and then generate a real value matrix via the Uniform random mechanism pertaining to the gapped histogram, derived in Step 2, for each entry's category accordingly and separately. Compute many copies of such mimicries and collectively call the collection the state ensemble Ω^a.

The binary slicing algorithm employed in Step 3 for generating a relative uniform block works reasonably well. Though it does not completely satisfy constraints of row and column sequences of empirical distributions, it comes very close. Therefore, the state ensemble Ω^a constructed here is the legitimate foundation for scientific inferences. In contrast, any generative algorithms developed for achieving the constraints of row and column sequences of sums is indeed lack of capability on matrix mimicking. A mimicking algorithm for binary directed networks is proposed and reported in a separate work [35].

3. APPLICATIONS

3.1. Assignment Problem

Consider a problem of assigning 40 tasks to 40 agents with a simulated cost matrix $\mathcal{M}^{AP}_0$. The design of this cost matrix reflects four levels of agent-task interactions or matching statuses: Excellent, Good, Average and Bad. That is, marginally there are four difficulty levels of tasks pertaining to an agent, and four skill levels of agents pertaining to a task. Specifically both 40 agents and 40 tasks are divided into eight core clusters of size five, and accordingly form eight "Excellent" relational 5×5 blocks located along the main diagonal. These eight blocks then couple with another eight "Easy" relational 5×5 blocks into four 10×10 blocks along the diagonal. These four blocks further coupled with another four "Average" relational 10×10 blocks into two 20×20 blocks also along the diagonal. The two off-diagonal 20×20 blocks are for "Bad" relational task-agent interaction.

By design these four scales, or levels of task-agent interacting block patterns constitute the deterministic structures embraced by $\mathcal{M}_0^{AP}$. That is, such deterministic structures specify marginal dependence on row and column via their Ultrametric trees, respectively, and task-agent dyadic dependence via multiscale blocks. From this dependence perspective an observed data matrix indeed embraces system characteristics. It also becomes clear that any statistical modeling upon matrix data, which is meant to coherently accommodate such complicate dependence based relational constraints, is a rather difficult undertaking, if not impossible.

Beyond the necessity of deterministic structural constraints, the system's true randomness is designed to equip with block-by-block uniformity and within-block stochasticity with possible complicate dependence constructs as well. That is, all costs within each relational block belonging to anyone of the four scales are collectively generated via an uniform stochastic mechanism. A general version can be in a form of discrete coupling measure mechanism [36], while the simplest version is the independently identically distributed (i.i.d.) mechanism. The discrete coupling measure mechanism is supposed to be better embracing general dependence patterns, but it is yet neither well known, nor well studied. On the other hand the i.i.d setting obviously captures the uniformity, but fail to bear with dependence of any realistic form within a block.

For illustrating purpose and simplicity, here we employ the block-wise uniformity under i.i.d setting of Poisson random variables with increasing intensity parameter values $\lambda = 1, 3, 5,$ or 7 along the four matching statuses from Excellent to Bad. A generated cost matrix, say $\mathcal{M}_0^{AP}$, is given in **Figure 1A**. It is evident that each block contains rather distinct degree of variation due to Poisson randomness. In order to accommodate such a volatile variation, the following distance is used in Data Mechanics computations: let $x, y \in Z^{40}$ be two nonnegative integer vectors,

$$d^2(x, y) = \sum_{i=1}^{40} \frac{(x_i - y_i)^2}{(x_i + y_i)/2}.$$

With such a distance measure as the initial empirical distance measure applied upon cost matrix $\mathcal{M}_0^{AP}$, our Data Mechanics algorithmic computations consequently and reliably recover the deterministic structures by design on $\mathcal{M}_0^{AP}$. And our block generative mechanism subject to constraints of row and column sequences of empirical distributions seems capable of nonparametrically capturing its inherent randomness without the prior knowledge of the original i.i.d. setting. We reiterate that there is no intention for recovering Poisson distributions in this Data Mechanics computational efforts. A state ensemble Ω_{AP} mimicking $\mathcal{M}_0^{AP}$ is also generated. Then the corresponding optimal assignment ensemble $\mathcal{B}^*_{AP}$ is constructed and collectively synthesized into a matrix $\Upsilon[\mathcal{B}^*_{AP}]$ via agents' assignments across all members of $\mathcal{B}^*_{AP}$, as shown in **Figure 1B**.

It is evident that the size of $\mathcal{B}^*_{AP}$ is much smaller than a given size of state ensemble, i.e., $|\mathcal{B}^*_{AP}| << |\Omega_{AP}|$. And it is seen that most of assignments are within the"excellent" scale of task-agent interaction. Thus, a series of 8 more or less block squares are clearly observed along the main diagonal of matrix $\Upsilon[\mathcal{B}^*_{AP}]$. But due to large variations inherited from Poisson randomness, some agents are indeed have larger collections of potential assignments than others. Since each optimal assignment in ensemble $\mathcal{B}^*_{AP}$ is equipped with a Boltzmann potential via $\Upsilon[\mathcal{B}^*_{AP}]$, the maximum potential assignment is naturally expected to be more systemic robust than the idiosyncratic optimal assignment based on the original cost matrix $\mathcal{M}_0^{AP}$. This is one difference that a system approach can make.

3.2. Traveling Salesman Problem and Simulated Annealing

Next we consider another well known discrete combinatorial optimization problem, traveling salesman problem (TSP). We consider simulated data via a design employed in the well known Simulated Annealing paper [16]. The whole set of involving city consists of 9 separately simulated collections. Each collection has 45 "cities" uniformly generated from a unit square. These 9 unit squares are arranged in 3×3 array on R^2. Upon these $405 (= 45 \times 9)$ cities, the observed data is the 405×405 distance matrix, $\mathcal{M}_0^{TSP}$. Unlike the above Assignment Problem, a computed Ultrametric tree is computed for both row and column axes via Data Mechanics. This ultrametric tree obviously provides only an unperfect approximation of Euclidean geometry of 405 points on R^2. However, its level of 9 clusters gives rise to a checkable "Guided Divide-and-Conquer (GDC)" feature: **the maximum distance of all cities to their individual nearest neighbor cities is much smaller than the minimum distances among the 9 squares.** Even though the naive greedy algorithm is not valid here, we develop the following divide-and-conquer scheme to show another significant merit of a system approach.

The significant merit is made possible based on "parallel computing" advantage offered by the GDC feature via the algorithm below:

GDC-1 Check the GDC feature on each levels of Ultrametric DCG-tree, and choose the level having largest number of clusters, say K. Upon such a coarse scale, we derive an $K \times K$ distance matrix by treating K clusters as K super-cities, and solve the corresponding optimal solution of TSP problem.

GDC-2 Along the optimal route through these K super-cities, find its entering- and leaving-city for each cluster.

GDC-3 Resolve an shortest route starting from the entering-city, and traveling over all other cities contained within the cluster and ending at the leaving-city.

GDC-4 Link the K optimal paths into one loop as the candidate optimal TSP solution.

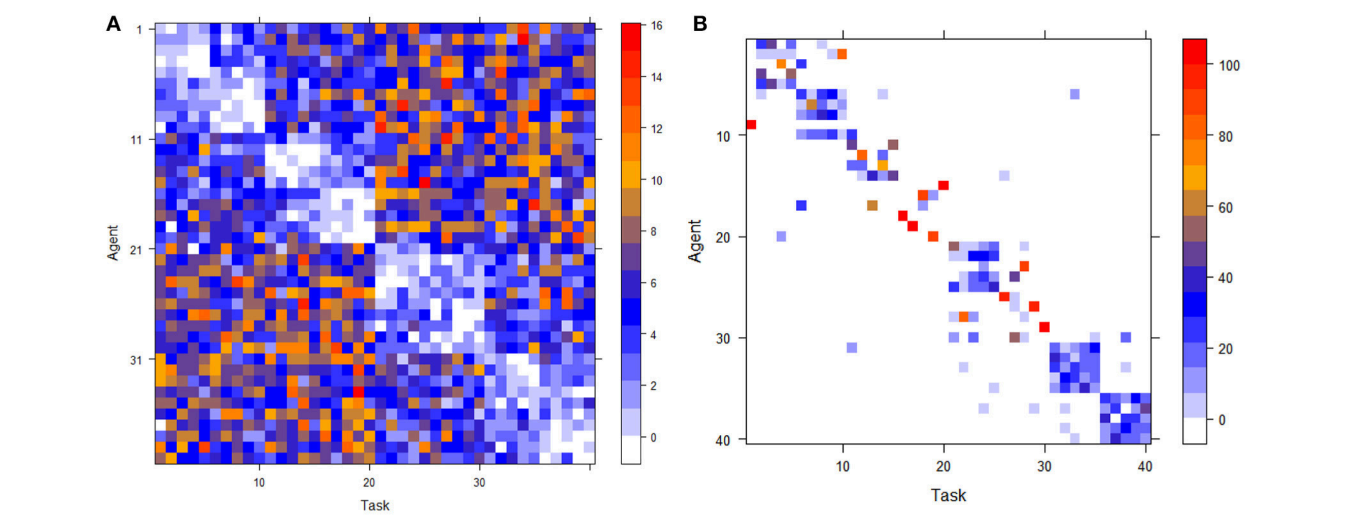

FIGURE 1 | Cost matrix (A) and Overall view of the ensemble of 100 optimal assignment solutions (B).

The division found in the GDC-1 step is crucially responsible for drastic reduction on computing complexity. And in GDC-3 step, this is exactly a much simpler permutation problem than TSP because of the tie-down constraint. Typically it can be resolved by applying Simulated Annealing algorithm. As illustrated, the Ultrametric tree in our coupling geometry based on $\mathcal{M}_0^{TSP}$ also satisfies the Guided Divide-and-Conquer feature on its 9-cluster level.

Again by applying the Data Mechanics, we build a state ensemble Ω_{TSP} of the city-system and repeatedly applying the GDC algorithm to build an ensemble $\mathcal{B}^*_{TSP}$ of optimal TSP solutions from each member of Ω_{TSP}. A view of successive cities connectivity through each optimal TSP solutions in ensemble $\mathcal{B}^*_{TSP}$ is collectively represented by a matrix $\Upsilon[\mathcal{B}^*_{TSP}]$, as shown in **Figure 2A** with its nine squares along the diagonal. That is, $\Upsilon[\mathcal{B}^*_{TSP}]$ reveals an aspect of rather restrictive local-connectivity in the geometry of $\mathcal{B}^*_{TSP}$.

One chief merit of ensemble $\mathcal{B}^*_{TSP}$ is that we can choose the optimal route in $\mathcal{B}^*_{AP}$ in the sense of achieving the minimum length upon $\mathcal{M}_0^{TSP}$. In our experiment with 1000 mimicries per cluster, this optimal route in $\mathcal{B}^*_{AP}$, by augmenting nine optimal sub-route from nine clusters, achieves a total length 11.51291 in comparison with the nearly true optimal route achieving 11.41388. These two versions of optimal route are indeed hardly distinguishable by raw eye sight, as shown in **Figure 2B** for the former version. Here the true optimal route is found by performing 1000 times of Simulated Annealing algorithm on each of the nine original clusters found on $\mathcal{M}_0^{TSP}$. This comparison reveals a fact that the state ensemble does provide a viable approximation with a computational edge. The state ensemble of Ω_{TSP} is also useful for deriving a new route when several cities are set to be skipped. An approximately optimal routes can be easily derived by "bridging-over" those skipped cities among all members of $\mathcal{B}^*_{TSP}$. Such a practical value of state ensemble of TSP is even more evident when the collection of cities is indeed very large. Since the possibility of obtaining the exact TSP solution becomes much smaller given a fixed budget of computing efforts. The concept of state ensemble offers other advantages as follows.

Another natural choice of TSP solution is the maximum Boltzmann potential route among $\mathcal{B}^*_{TSP}$ based on the potentials recorded in $\Upsilon[\mathcal{B}^*_{TSP}]$. This choice is likely more robust when distance measurements in matrix $\mathcal{M}_0^{TSP}$ are subject to measurement errors. The reason is that some averaging mechanism has been implicitly built into $\Upsilon[\mathcal{B}^*_{TSP}]$. One further merit of state ensemble is going against the folklore practice in general optimization algorithms, including Simulated Annealing and Genetical Algorithm. Researchers typically adopt an common remedial practice by blindly starting an optimization algorithm from as many different initial locations as possible, and then choose the best route. Through the sharp contrasting view of geometry revealed in **Figure 2A**, we are almost certain that this practice surely would incur huge waste in computing because their random trajectories mount to wander too widely and stray too far away from any reasonable solution routes with large probabilities.

4. DISCUSSION

We demonstrate a machine learning algorithm, Data Mechanics, that can compute a geometry coupling deterministic multiscale blocks patterns with stochastic block-wise uniform randomness on a data matrix. When a discrete combinatorial optimization problem is defined upon a data matrix representing a system at one point in time, such a coupling geometry allows us to go beyond the classic idiosyncratic optimal solution. The ensemble of optimal solutions from the ensemble of system's microstates make possible for us to look into robustness/sensitivity issue, and at the same time to propose

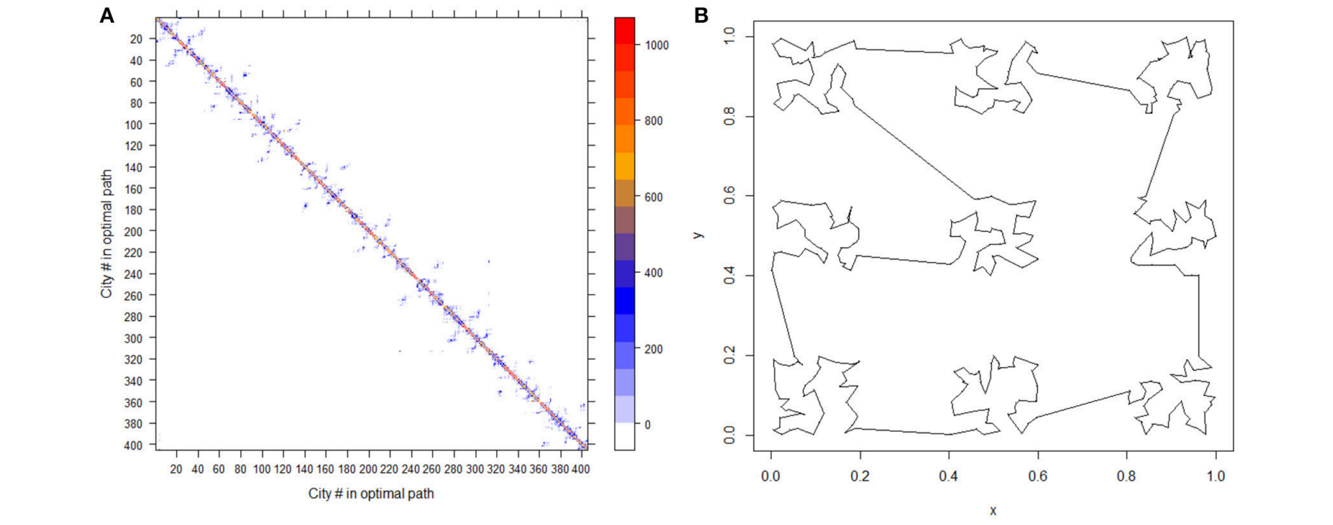

FIGURE 2 | An overall view of the ensemble of 1000 optimal TSP solutions (A) and the optimal TSP solution (B).

robust modifications via the concept of maximum Boltzmann potential.

The Assignment Problem illustrates the pertinent interacting patterns on a product node spaces of subjects and tasks, while the Traveling Salesman Problem brings out the intrinsic geographic geometry among all "cities." Such data-driven pattern information turns out to be essential for resolving discrete mathematical and computer science problems not only from the algorithmic computing perspective, but also from the perspective of having meaningful real world systemic explanations. Indeed such considerations might be necessary when the system under study is complex.

From a technical aspect, it is noted that there is still a large room for improvements upon how to mimic real-value or categorical matrices. In fact, even for a large binary matrix, the current available algorithms still fall short of providing reasonable results. From this aspect, the Data Mechanics provides an imperative advantage by discovering a coupling geometry on a big matrix, so that the mimicking task can be divided-and-conquered.

AUTHOR CONTRIBUTIONS

FH, initiated the research problem, design computer experiments, data analysis, writing. KF, designed and carried out computer experiments and data analysis. CH, discussed the general issues and related topics.

REFERENCES

1. Kuhn HW. The Hungarian method for the assignment problem. *Naval Res Logist Q.* (1955) **2**:83–97. doi: 10.1002/nav.3800020109
2. Munkres J. Algorithms for the assignment and transportation problems. *J Soc Indust Appl Math.* (1957) **5**:32–8. doi: 10.1137/0105003
3. Burkard R, Dell'Amico M, Martello S. *Assignment Problems (Revised reprint).* Minneapolis, MN: SIAM (2012). doi: 10.1137/1.9781611972238
4. Bellman R. Dynamic programming treatment of the travelling salesman problem. *J Assoc Comput Mach.* (1962) **9**:61–3. doi: 10.1145/321105.321111
5. Papadimitriou CH. The euclidean traveling salesman problem is NP-complete. *Theor Comput Sci.* (1977) **4**:237–44. doi: 10.1016/0304-3975(77)90012-3
6. Arora S. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J ACM* (1998) **45**:753–82. doi: 10.1145/290179.290180
7. Lawler EL, Lenstra JK, Rinnooy KAHG, Shmoys DB. *The Traveling Salesman Problem.* New York, NY: Wiley (1985).
8. Applegate DL, Bixby RM, Chvátal V, Cook WJ. *The Traveling Salesman Problem.* Princeton, NJ: Princeton Univesity Press. (2006).
9. Diaconis P. Patterns in eigenvalues: the 70th Josiah Willard Gibbs Lecture. *Bull Amer Math Soc.* (2003) **40**:155–78. doi: 10.1090/S0273-0979-03-00975-3
10. Forrester P, Snaith N, Verbaarschot V. Introduction in review to special issue on random matrix theory. *J Phys A Math Gen.* (2003) **36**:R1–10. doi: 10.1088/0305-4470/36/12/201
11. Mehta M, *Random Matrices. 3rd Edn.* Amsterdam: Elsevier (2004).
12. Mezard M, Parisi G, Virasoro M. *Spin Glass Theory and Beyond.* Singapore: World Scientific (1986). doi: 10.1142/0271
13. Parisi G. A conjecture on random bipartite matching. arXiv:cond-mat/9801176v1 (1998).
14. Krokhmal PA, Pardalos PM. Random assignment problems. *Eur J Oper Res.* (2009) **194**:1–17. doi: 10.1016/j.ejor.2007.11.062
15. Kapp RM. Probabilistic analysis of partitioning algoritms for the TSP in the plane. *Math Oper. Res.* (1977) **2**:209–24. doi: 10.1287/moor.2.3.209
16. Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simulated annealing. *Science* (1983) **220**:671–80. doi: 10.1126/science.220.4598.671
17. Erdos P, Rényi A. On random graphs I. *Publ Math Debrecen* (1959) **6**:290–7.
18. Bollobás B. *Random Graphs, 2nd Edn.* Cambridge: Cambridge University Press (2001). doi: 10.1017/CBO9780511814068
19. Dyson FJ. Statistical theory of the energy levels of complex systems-I. *J Math Phys.* (1962) **3**:140. doi: 10.1063/1.1703773
20. Fushing H, McAssey MP. Time, temperature and data cloud geometry. *Phys Rev E* (2010) **82**:061110. doi: 10.1103/PhysRevE.82.061110

21. Fushing H, Wang H, Van der Waal K, McCowan B, Koehl P. Multi-scale clustering by building a robust and self-correcting ultrametric topology on data points. *PLoS ONE* (2013) **8**:e56259. doi: 10.1371/journal.pone.0056259
22. Fushing H, Chen C. Data mechanics and coupling geometry on binary bipartite network. *PLoS ONE* (2014) **9**:e106154. doi: 10.1371/journal.pone.0106154
23. Fushing H, Hsueh CH, Heitkamp C, Matthews M, Koehl P. Unravelling the geometry of data matrices: effects of water stress regimes on winemaking. *J R Soc Interface* (2015). **12**:20150753. doi: 10.1371/journal.pone.0106154
24. Crutchfield JP. Between order and chaos. *Nat Phys*. (2012) **8**:17–24. doi: 10.1038/nphys2190
25. Rissanen J. *Stochastic Complexity and Statistical Inquiry*. Singapore: World Scientific (1989).
26. Li M, Vitányi P. *An Introduction to Kolmogorov Complexity and its Applications*. New York, NY: Springer-Verlag (1993).
27. Hsieh CJ, Si S, Dhillon IS. A divide-and-conquer solver for kernel support vector machines. In: *International Conference on Machine Learning*, Beijing. (2014).
28. Hsieh CJ, Banerjee A, Dhillon IS, Ravikumar P. A divide-and-conquer method for sparse inverse covariance estimation. In: *Advances in Neural Information Processing Systems*, Lake Tahoe. (2012).
29. Si S, Shin D, Dhillon IS, Parlett BN. Multi-scale spectral decomposition of massive graphs. In: *Advances in Neural Information Processing Systems*, Montreal, QC. (2014).
30. Mackey LW, Jordan MI, Talwalkar A. Divide-and-conquer matrix factorization. In: *Advances in Neural Information Processing Systems*, Granada. (2011).
31. Bayati M, Kim JH, Saberi A. A sequential algorithm for generating random graphs. *Algorithmica* (2010) **58**:860–910. doi: 10.1007/s00453-009-9340-1
32. Chen Y, Diaconis P, Susan PH, Liu JS. Sequential Monte Carlo methods for statistical analysis of tables. *J Am Stat Assoc.* (2005) **100**:109–20. doi: 10.1198/016214504000001303
33. Barvinok A. What does a random contingency table look like? *Combinator Probab Comput.* (2010) **19**:517–39. doi: 10.1017/S0963548310000039
34. Fushing H, Chen C, Liu SY, Koehl P. Bootstrapping on undirected binary network via statistical mechanics. *J Stat Phys.* (2014) **156**:823–42. doi: 10.1007/s10955-014-1043-6
35. Fushing H, Fujii K. Mimicking directed binary networks for exploring systemic sensitivity: Is NCAA FBS a fragile competition system? *Front Appl Math Stat.* (2016). **2**:9. doi: 10.3389/fams.2016.00009
36. Mémoli F. Spectral Gromov-Wasserstein distances for shape matching. In: *Proc ICCV Workshops* (2009). p. 256–63. doi: 10.1109/iccvw.2009.5457690

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.