JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 87, No. 1, pp. 197-220, OCTOBER 1995

Matrix Representation and Gradient Flows for NP-Hard Problems

W. S. WONG[1]

Communicated by W. B. Gong

Abstract. Over the past decade, a number of connections between continuous flows and numerical algorithms were established. Recently, Brockett and Wong reported a connection between gradient flows on the special orthogonal group $\mathcal{SO}(n)$ and local search solutions for the assignment problem. In this paper, we describe a uniform formulation for certain NP-hard combinatorial optimization problems in matrix form and examine their connection with gradient flows on $\mathcal{SO}(n)$. For these problems, there is a correspondence between the so-called 2-opt solutions and asymptotically stable critical points of an associated gradient flow.

Key Words. Gradient flows, assignment problem, traveling salesman problem, graph partitioning problem, local search.

1. Introduction

Over the past decade, a number of connections between continuous flows and numerical algorithms were established; see, for example, Bayer and Lagarias (Ref. 1), Bloch (Ref. 2), Brockett (Refs. 3 and 4), Brockett and Wong (Ref. 5), Chu (Ref. 6), Deift, Nanda, and Tomei (Ref. 7), Faybusovich (Refs. 8 and 9), Symes (Refs. 10 and 11), and Watkins and Elsner (Ref. 12). Included in this list, among many others, are the classical connection between the Toda lattice flow and the QR algorithm, the relation of affine and projective trajectories with the Karmarkar algorithm and its variants, and the connection between gradient flows on the special orthogonal group and the least squares optimization problem.

These results are of interest to a variety of audiences. From the system theory perspective, other than their potential of leading to advances in

[1]Reader, Department of Information Engineering, Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.

computation, they provide many novel classes of dynamic systems with interesting structures and properties, such as the so-called double bracket equations introduced in Ref. 3.

In this paper, we will describe a uniform formulation for certain NP-hard combinatorial optimization problems in matrix form and examine their connection with gradient flows on the space $\mathcal{SO}(n)$ of the special orthogonal matrices.

In Ref 13, Karmarkar proposed using a steepest descent approach to solve combinatorial optimization problems; see also Refs. 14–16. Independently in Ref. 5, the gradient flow approach was applied to a class of combinatorial optimization problems known as the assignment problem. One of the key results reported in that paper is that, for any given assignment problem, there is a simple correspondence between local minima of a certain local search algorithm and local minima of an associated gradient flow defined on $\mathcal{SO}(n)$.

Since the assignment problem is known to have a polynomial time solution (Ref. 17), an algorithm to find a local minimum does not have great significance. In this paper, however, we show that the $\mathcal{SO}(n)$ gradient flow approach can be extended easily to a large class of combinatorial optimization problems that includes the traveling salesman problem and the graph partitioning problem. Since these problems are well known to be NP-hard, a local search algorithm is commonly used to solve approximately these problems. Thus, results connecting local search algorithms for these problems with gradient flows may have a more practical significance. Moreover, the problem of finding a local minimum may be computationally complex itself. The complexity issue of local search algorithms is an interesting topic first raised by Johnson, Papadimitriou, and Yannakakis (Ref. 18). In particular, for the graph partitioning problem with the SWAP neighborhood (two partitions are neighbors if one can be made identical to the other by swapping two vertices), it was shown by Schäffer and Yannakakis (Ref. 19) that finding the local minimum from an arbitrary initial point is NP-hard. One of our theorems here shows that the set of local minima for the graph partitioning problem with the SWAP neighborhood has a simple correspondence with the asymptotically stable critical points of an associated gradient flow on $\mathcal{SO}(n)$.

The extension of the gradient flow approach to these NP-hard problems is based on the observation that any combinatorial optimization problem is representable as an optimization problem on the set of permutation matrices. Such a matrix form representation for many of the commonly known combinatorial optimization problems takes on a very simple form, and in the case of the traveling salesman problem is probably well known in the folklore. In the first part of this paper, we will present a general discussion of the representation of combinatorial optimization problems in this matrix form.

In Section 3, we describe basic concepts of local search algorithms and show how various combinatorial problems can be embedded as continuous optimization problems on $\mathscr{SO}(n)$. In Section 4, we prove a theorem connecting local search minima with local minima of associated gradient flows on $\mathscr{SO}(n)$.

2. Matrix Representation of Combinatorial Optimization Problems

Let $\mathscr{S}_n$ represent the symmetric group on n symbols. By an n-symbol combinatorial optimization problem, we mean the problem of finding a globally optimal value of an arbitrary function defined on $\mathscr{S}_n$. Without loss of generality, we deal with minimization problems only in this paper. Let $\mathscr{P}_n$ represent the set of $n \times n$ permutation matrices; there is a one-to-one, onto correspondence between elements in $\mathscr{S}_n$ and $\mathscr{P}_n$. By viewing $\mathscr{P}_n$ as an incidence matrix for a graph, one also obtains a one-to-one, onto correspondence between $\mathscr{P}_n$ and $\mathscr{G}_n$, the set of directed graphs with the property that every vertex is the source of exactly one directed edge and the sink of exactly one directed edge. We use the convention that P_{ij} is equal to 1 if and only if there is a directed edge from vertex i to vertex j; i is called the source and j is the sink of the directed edge.

Let $\operatorname{tr}(M)$ represent the trace of matrix M. If C is the cost matrix defining an assignment problem, it is well known that it can be represented in the form

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}(C^T P). \tag{1}$$

It is possible to extend this representation to other combinatorial optimization problems. In fact, the following result holds.

Proposition 2.1. Any n-symbol combinatorial optimization problem can be represented in the form

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}\left(\sum_{i_1, i_2, \ldots, i_n} D_{i_1} P D_{i_2} P \cdots D_{i_n} P \right), \tag{2}$$

where the D_i are $n \times n$ matrices of rank 1.

Proof. It is sufficient to show that, for any permutation matrix P_0, there exists a function $\xi(P)$ of the form $\operatorname{tr}(D_1 P D_2 P \cdots D_n P)$ such that

$$\xi(P) = \begin{cases} 1, & \text{if } P = P_0, \\ 0, & \text{otherwise.} \end{cases}$$

Denote by $\vec{1}_i$ the column vector with all entries zero, except at the i position, where the entry is equal to 1. Define

$$\vec{1}_{\sigma(i)} = P_0 \vec{1}_i.$$

It is easy to see that a permutation matrix P is equal to P_0 if and only if $\vec{1}^T_{\sigma(i)} \mathrm{P} \vec{1}_i$ is nonzero and equal to 1 for all $1 \leq i \leq n$. Since

$$\vec{1}^T_{\sigma(1)} P \vec{1}_1 \vec{1}^T_{\sigma(2)} P \vec{1}_2 \cdots \vec{1}^T_{\sigma(n)} P \vec{1}_n = \operatorname{tr}(\vec{1}_n \vec{1}^T_{\sigma(1)} P \vec{1}_1 \vec{1}^T_{\sigma(2)} P \vec{1}_2 \cdots \vec{1}^T_{\sigma(n)} P),$$

it follows that, by defining

$$D_1 = \vec{1}_n \vec{1}^T_{\sigma(1)}, \qquad D_i = \vec{1}_{i-1} \vec{1}^T_{\sigma(i)}, \quad \text{for } i > 1,$$

ξ will have the desired property. □

Since the matrices D_i are of rank 1, it follows that the expression in (2) can also be expressed as

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}\left(\sum_{i_1, i_2, \ldots, i_n} D_{i_1} P^T D_{i_2} P \cdots D_{i_n} P \right). \tag{3}$$

For commonly encountered problems, the expression in Eq. (2) or (3) can usually be simplified. We consider three classes of examples.

Example 2.1. Assignment Problem. It was shown in Ref. 5 that the assignment problem can be defined in the form

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}\left(\sum_{i=1}^{r} X_i P^T Y_i P \right), \tag{4}$$

where r is the rank of the cost matrix C and where X_i and Y_i are diagonal matrices obtained by the following outer-product decomposition of C:

$$C = \sum_{i=1}^{r} X_i \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} [1, \ldots, 1] Y_i. \tag{5}$$

For a detailed study of the relation between the assignment problem and its associated gradient flow, see Ref. 5.

Example 2.2. Traveling Salesman Problem. The traveling salesman problem is similar to the assignment problem, except that the optimization is restricted to n-cycles only. In the graphical representation, these permutations correspond to directed circuits with n-vertices. Although, strictly speaking, this is not a problem defined on $\mathscr{S}_n$, we can treat the traveling salesman problem as an n-symbol combinatorial optimization problem by making use of the observation that the set of permutation matrices corresponding to all n-cycles is equal to the set of the matrices of the form $P^T S_{\mathscr{T}}(n) P$, where P is a permutation matrix and

$$S_{\mathscr{T}}(n) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Hence, the traveling salesman problem can be formulated as

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}(C^T P^T S_{\mathscr{T}}(n) P). \tag{6}$$

Notice that, in this formulation, different elements in $\mathscr{S}_n$ may correspond to the same directed circuit.

One can also extend this representation to other variants of the traveling salesman problem. As an example, consider a two traveling salesmen problem, with the restriction that p cities are toured by one salesman and q cities by the other, with $n = p + q$. We can formulate this problem as

$$\min_{P \in \mathscr{P}_n} \operatorname{tr}(C^T P^T S_{\mathscr{G}\mathscr{T}}(n) P), \tag{7}$$

where

$$S_{\mathscr{G}\mathscr{T}}(n) = \begin{bmatrix} S_{\mathscr{T}}(p) & 0_{pq} \\ 0_{qp} & S_{\mathscr{T}}(q) \end{bmatrix},$$

and where 0_{pq} is the $p \times q$ matrix with all zero entries.

Example 2.3. Graph Partitioning Problem. Let G be a fully connected undirected graph with n vertices with weights assigned to its undirected edges. Let p and q be two positive integers such that their sum is equal to n. The generalized graph partitioning problem is to find a partition of vertices into two subsets with p and q elements such that the sum of the weights on the cut edges (that is, edges with their endpoints in different subsets of the partition) is minimized. The case where p and q are equal defines the classic graph partitioning problem. The graph partitioning problem is known to be NP-hard.

Just like the traveling salesman problem, the graph partitioning problem is not defined on $\mathcal{S}_n$, strictly speaking. However, we can formulate a corresponding problem on $\mathcal{S}_n$ by first constructing a fully connected directed graph with n vertices. For any original weight assigned to an undirected edge, we assign half of it to each of the two directed edges joining the same vertices in the undirected graph. Call the cost matrix corresponding to this new set of weights C. Now, define

$$S_{\mathcal{G}}(n)=\begin{bmatrix} 0_{pp} & 1_{pq} \\ 1_{qp} & 0_{qq} \end{bmatrix}, \tag{8}$$

where 1_{ij} denote the $i\times j$ matrix with all entries equal to 1. Then, the graph partitioning problem can be represented in the following way:

$$\min_{P\in\mathcal{P}_n} \operatorname{tr}(C^TP^TS_{\mathcal{G}}(n)P). \tag{9}$$

It is possible to extend the graph partitioning problem to a problem of partitioning into k subsets, where the subsets have cardinalities $n_1,\dots,n_k$, forming a partition of the integer n. In this case, the formulation in (9) still holds, provided $S_{\mathcal{G}}(n)$ is now replaced by

$$S_{\mathcal{G}\mathcal{G}}(n)=1_{nn}-\begin{bmatrix} 1_{n_1n_1} & 0_{n_1n_2} & \cdots & 0_{n_1n_k} \\ 0_{n_2n_1} & 1_{n_2n_2} & \cdots & 0_{n_2n_k} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0_{n_kn_1} & 0_{n_kn_2} & \cdots & 1_{n_kn_k} \end{bmatrix}.$$

3. Local Search and 2-Change Neighborhood

For the rest of this paper, we will consider combinatorial optimization problems of the form

$$\min_{P\in\mathcal{P}_n} \operatorname{tr}\left(\sum_{i=1}^{r} X_i^TP^TY_iP\right), \tag{10}$$

where X_i and Y_i are arbitrary matrices. Thus, this class of problems contains all three examples presented in the previous section.

Local search algorithms are commonly used to solve NP-hard problems. All these algorithms require the definition of a neighborhood around an arbitrary element in the search space. The k-change neighborhood is a popular concept that is also known to be quite efficient for the traveling salesman problem and the graph partitioning problem.

For any n-symbol combinatorial optimization problem, we can define a k-change neighborhood in the following way. For any matrix $\mathscr{P}$ in P_n, let G be the corresponding directed graph in $\mathscr{G}_n$. The k-change neighborhood of G, and hence P, is then defined as the set of elements in $\mathscr{G}_n$ that can be obtained by removing k directed edges from G and then placing k alternative directed edges to the remaining graph. For example, the permutation matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

has six neighbors in the 2-change neighborhood:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

With this motivation, we define the 2-change neighborhood of P, where P is an element in $\mathscr{P}_n$, to consist of the $n(n-1)/2$ permutation matrices of the form PN_{ij}, where N_{ij} is a permutation matrix that has only two nonzero off diagonal entries, at (i, j) and (j, i). With this as motivation, we give formally the following definition.

Definition 3.1. For the combinatorial optimization problem defined by (10), the 2-change neighborhood of a permutation matrix P consists of P and matrices of the form PN_{ij}, for $1 \leq i < j \leq n$. An element of the form

PN_{ij}, where

$$(PN_{ij})^T Y_k PN_{ij} \neq Y_k,$$

for some k, $1 \leq k \leq r$, is called a distinct neighbor of P.

Once a neighborhood is chosen, a local search algorithm imitates essentially a gradient descent algorithm, although the rules may vary if there is more than one possible choice to improve the current solution. Since these details will not affect our discussion, we will simply assume that a fixed local search algorithm has been chosen from now on.

Definition 3.2. A 2-opt solution for the combinatorial optimization problem defined by (10) is defined as a locally minimal solution obtained under the chosen local search algorithm using the 2-change neighborhood. The solution is called nondegenerate if its value is strictly lower then the value at its distinct neighbors.

Notice that the definition of a 2-change neighborhood is independent of the parameters of the problem, but the definition of a distinct neighbor is dependent on the Y_k. The motivation for this will become clear when we study the graph partitioning problem and the traveling salesman problem.

To illustrate the definition of a 2-change neighborhood, let us apply it to a graph partitioning problem with $p=q=2$. Recall that the cost function for this problem is formulated as $\mathrm{tr}(C^T P^T S_{\mathscr{G}}(n) P)$, with the state of the partition represented by $P^T S_{\mathscr{G}}(n) P$. For simplicity, consider the neighborhood around the identity matrix I_n. The matrix

$$I^T S_{\mathscr{G}}(n) I = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

represents the state that divides the graph into two subgraphs, one consisting of vertices 1 and 2, the other consisting of vertices 3 and 4. The matrix $N_{13}^T S_{\mathscr{G}}(n) N_{13}$ is given by

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

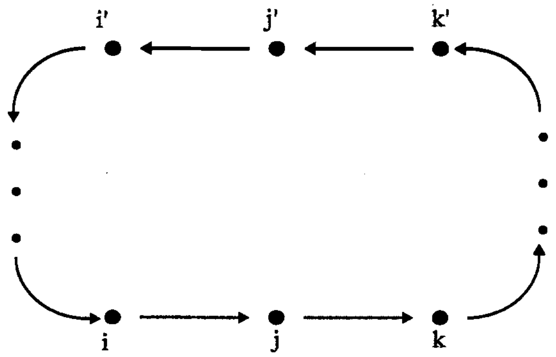

Fig. 1. Directed circuit θ.

This corresponds to the partition that groups vertex 1 with 4 and vertex 2 with 3. In other words, we have swapped vertex 1 with vertex 3. Since vertex 1 and vertex 2 are in the same subset in the original partition, swapping the two vertices should produce the same partition. This is confirmed by the fact that

$$N_{12}^T S_{\mathscr{G}} N_{12} = S_{\mathscr{G}}.$$

According to our definition, N_{12} is not a distinct neighbor of the identity matrix. Generalizing this argument, one can show that this definition of the 2-change neighborhood coincides with the definition of the SWAP neighborhood. Moreover, a nondegenerate 2-opt solution defined here is a nondegenerate 2-opt solution in the classical sense.

Applying this neighborhood concept to the traveling salesman problem leads to two types of new circuits as illustrated in Figs. 1–3. It is clear that this definition of the neighborhood is related but not identical to the classical

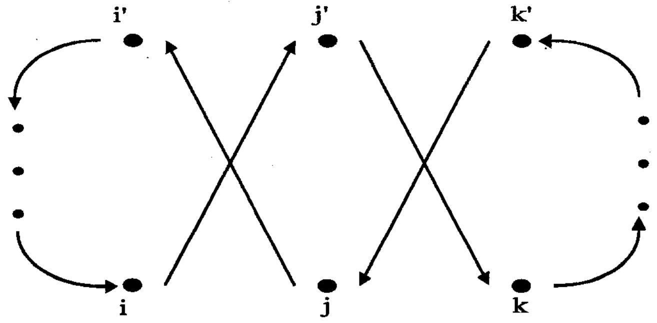

Fig. 2. Directed circuit corresponding to $\theta N_{jj'}$, when vertex j and vertex j' are not directly connected.

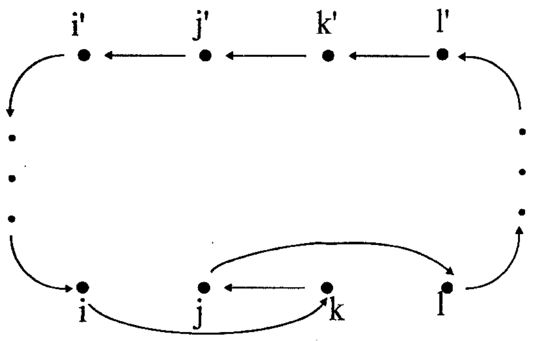

Fig. 3. Directed circuit corresponding to θN_{jk} when there is an edge from vertex j to vertex k.

definition of a 2-opt neighborhood for the traveling salesman problem. The difference comes from the fact that the neighborhood defined here is obtained by interchanging vertices rather than edges.

4. Embedding in $\mathscr{SO}(n)$ and Equivalence of Optimality

Since $\mathscr{P}_n$ is a subset of $\mathscr{O}(n)$, a subgroup in fact, one can embed the optimization problem defined by (10) to an optimization problem defined in $\mathscr{O}(n)$, namely,

$$\min_{\Theta \in \mathscr{SO}(n)} \operatorname{tr}\left(\sum_{i=1}^{r} X_i^T \Theta^T Y_i \Theta \right). \tag{11}$$

For the case where X_i and Y_i are diagonal matrices, this type of optimization problem, restricted to the connected component $\mathscr{SO}(n)$, containing the identity matrix, was considered in detail in Ref. 3 and was shown to be related to solving geometric matching problems.

For cases where the X_i and Y_i are not all diagonal, the optimization problem defined in (11) has one undesirable feature. Namely, since $\mathscr{O}(n)$ contains elements of the form DP the so-called hyperoctahedral matrices, where D is a diagonal matrix with diagonal values 1 or -1, this embedding may introduce many spurious local minima. In order to eliminate these spurious local minima, one can define a different embedded problem by defining a cost function η on $\mathscr{O}(n)$ by

$$\eta(\Theta) = \operatorname{tr}\left(\sum_{i=1}^{r} X_i^T (\Theta \circ \Theta)^T Y_i \Theta \circ \Theta \right), \tag{12}$$

where $M \circ N$ denote the Schur–Hadamard product of two matrices defined by $(M_{ij}N_{ij})$. If we identify elements in $\mathscr{P}_n$ with elements in $\mathscr{SO}(n)$ by the following mapping:

$$\iota(P) = \begin{cases} P, & \text{if } \det(P) = 1, \\ \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 1 \end{bmatrix} P, & \text{if } \det(P) = -1, \end{cases}$$

then one can show that

$$\eta(\iota(P)) = \operatorname{tr}\left(\sum_{i=1}^{r} X_i^T P^T Y_i P\right). \tag{13}$$

So, η preserves the values of the cost function at all permutation matrices, and it is sufficient to consider the optimization problem on $\mathscr{SO}(n)$. From now on, we will consider an embedded optimization problem of the form

$$\min_{\Theta \in \mathscr{SO}(n)} \operatorname{tr}(\eta(\Theta)). \tag{14}$$

Since the inverse of ι is given by

$$\iota^{-1}(H) = H \circ H,$$

we can extend the definition of ι^{-1} to all hyperoctahedral matrices. Notice that different local minima of η may correspond to the same element in $\mathscr{P}_n$.

Although the value of the cost function is preserved in this setting, there still remains the important question of whether an optimal solution in the original combinatorial optimization problem is preserved as a global optimum for the problem defined in (14). The answer to this question is positive for the least square matching problem and the assignment problem. It is not known to hold for a problem defined by (10) in general. A more realistic requirement is that the local optimality structure be preserved.

Since η can be viewed as a potential function on a compact Riemannian manifold, it is natural to consider the gradient flow it induces. It is well known that $\mathscr{SO}(n)$ is a Lie group and the tangent space at any of its element can be identified with the space of antisymmetric matrices $o(n)$. Define an inner product on $o(n)$ by

$$\langle \Pi, \Omega \rangle = \operatorname{tr}(\Pi^T \Omega).$$

This inner product defines a Riemannian structure on $\mathscr{SO}(n)$.

It is natural to study the gradient flow induced by η under this standard Riemannian structure. Moreover, if the embedding defined previously preserves the local optimality structure, we may expect that there should be a correspondence between asymptotically stable critical points of this induced gradient flow and 2-opt solutions. The main result of this paper is to prove that this correspondence holds for certain combinatorial optimization problems.

Denote the gradient flow induced by η by

$$d\Theta/dt = -\nabla\eta(\Theta(t)). \tag{15}$$

Strictly speaking, Eq. (15) should be interpreted as a dynamical equation on $\mathcal{SO}(n)$. However, since $\mathcal{SO}(n)$ is embedded in the space of all $n \times n$ matrices, one can view Eq. (15) as a matrix differential equation with the understanding that, if the system starts at a point in $\mathcal{SO}(n)$, its trajectory for all time remains in $\mathcal{SO}(n)$. As observed by Faybusovich (Refs. 8–9), this equation can be integrated numerically without requiring any matrix inversion, unlike many other continuous flow algorithms.

Ideally, for a given combinatorial optimization problem, we would like to find an embedding of the problem into a dynamical system so that there is a one-to-one, onto correspondence between locally optimal solutions, say 2-opt solutions, and asymptotically stable critical points of the dynamical system. If this is true, then the combinatorial optimization problem is equivalent to the dynamical system, in the sense that one can develop an equivalent, analog approach to find 2-opt solutions. It turns out that, for the $\mathcal{SO}(n)$ embedding, it is difficult to guarantee that the correspondence is onto. Hence, we introduce a weaker concept of equivalence.

Definition 4.1. The weak gradient flow equivalence property is said to hold for a combinatorial optimization problem defined in (10) if:

(i) the image of every nondegenerate 2-opt solution under ι is an asymptotically stable critical point for the gradient flow system defined by Eq. (15);

(ii) for every H, a hyperoctahedral, asymptotically stable critical point of the gradient flow system defined by Eq. (15), $\iota^{-1}(H)$ is a 2-opt solution.

We call this weak equivalence because condition (ii) allows the possibility that some asymptotically stable critical points may have no corresponding 2-opt solutions. Hence, if one uses the embedding dynamical system to find 2-opt solutions, one may end up with nonmeaningful solutions. As discussed in the numerical study section (Section 8), these nonmeaningful solutions may yield nevertheless useful heuristic solutions.

Our main results in this paper show that, for the graph partitioning problem and some cases of the traveling salesman problem, the weak gradient flow equivalence property holds. As a consequence, one may be able to employ the gradient flow approach to find suboptimal solutions to these problems, which are in fact locally optimal if they correspond to meaningful solutions.

5. Conditions for Weak Gradient Equivalence

For any $1 \le i < j \le n$, define the matrices $\Omega(i,j)$ by

$$\Omega(i,j)_{pq} = \begin{cases} 1, & \text{if } i=p, j=q, \\ -1, & \text{if } i=q, j=p, \\ 0, & \text{otherwise.} \end{cases}$$

The matrices $\Omega(i,j)$ form a basis for $o(n)$. Define

$$\Lambda(i,j)_{pq} = \begin{cases} 1, & \text{if } p=q=i, \\ 1, & \text{if } p=q=j, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\Pi(i,j) = \Omega(i,j) \circ \Omega(i,j) - \Lambda(i,j).$$

So, $\Pi(i,j)$ is the symmetric matrix with -1 at entries (i,i) and (j,j), $+1$ at entries (i,j) and (j,i), and 0 elsewhere.

Theorem 5.1. The weak gradient flow equivalence property holds for the combinatorial optimization problem defined by (10) if and only if, at any hyperoctahedral critical point Θ of Eq. (15), the following property holds for all $1 \le i < j \le n$:

if $(\eta(\Theta N_{ij}) - \eta(\Theta)) > 0$,

$$\text{then } \left(\eta(\Theta N_{ij}) - \eta(\Theta) - \operatorname{tr}\left(\sum_{k=1}^{r} X_k^T (\Theta \Pi(i,j))^T Y_k \Theta \Pi(i,j)\right)\right) > 0; \tag{16a}$$

$$\text{if } \left(\eta(\Theta N_{ij}) - \eta(\Theta) - \operatorname{tr}\left(\sum_{k=1}^{r} X_k^T (\Theta \Pi(i,j))^T Y_k \Theta \Pi(i,j)\right)\right) \ge 0,$$

$$\text{then } (\eta(\Theta N_{ij}) - \eta(\Theta)) \ge 0. \tag{16b}$$

The crucial step in proving Theorem 5.1 is to analyze $H_\eta(\Theta)$, the Hessian of η at a hyperoctahedral Θ. To compute the Hessian, instead of using brute force, we will compute it through a series of simple lemmas.

Lemma 5.1. If Θ is hyperoctahedral and A is arbitrary, then

$$(\Theta A)\circ(\Theta A)=(\Theta\circ\Theta)(A\circ A).$$

Proof. The (i,j) entry of $(\Theta A)\circ(\Theta A)$ is $(\sum_{k=1}^n \Theta_{ik}A_{kj})^2$. Since Θ is hyperoctahedral, the sum has only one nonzero term. Hence,

$$\left(\sum_{k=1}^n \Theta_{ik}A_{kj}\right)^2=\sum_{k=1}^n \Theta_{ik}^2A_{kj}^2,$$

which is the (i,j) entry of $(\Theta\circ\Theta)(A\circ A)$. □

Parametrize a neighborhood of an arbitrary point Θ in $\mathcal{SO}(n)$ by

$$\Theta(\Omega)=\Theta(1+\Omega+\Omega^2/2!+\cdots). \tag{17}$$

To compute the gradient of η at Θ, it is sufficient to consider the restriction of η to the $n(n-1)/2$ one-parameter subgroups of the form

$$\begin{aligned}a_{ij}(t,\Theta)&=\eta(\Theta\, e^{t\Omega(i,j)})\\&=\mathrm{tr}\left(\sum_{i=1}^r X_i^T((\Theta\, e^{t\Omega(i,j)})\circ(\Theta\, e^{t\Omega(i,j)}))^TY_i((\Theta\, e^{t\Omega(i,j)})\circ(\Theta\, e^{t\Omega(i,j)}))\right).\end{aligned} \tag{18}$$

For $1\le k\le n$, define

$$\tilde{Y}_k(\Theta)=(\Theta\circ\Theta)^TY_k(\Theta\circ\Theta).$$

Then by Lemma 5.1, if Θ is hyperoctahedral,

$$a_{i,j}(t,\Theta)=\mathrm{tr}\left(\sum_{i=1}^r X_i^T(e^{t\Omega(i,j)}\circ e^{t\Omega(i,j)})^T\tilde{Y}_i(\Theta)(e^{t\Omega(i,j)}\circ e^{t\Omega(i,j)})\right). \tag{19}$$

Since

$$\Omega(i,j)^2=-\Lambda(i,j)\quad\text{and}\quad\Omega(i,j)^3=-\Omega(i,j),$$

it follows that

$$\begin{aligned}e^{t\Omega(i,j)}&=I_n+(t-t^3/3!+t^5/5!+\cdots)\Omega(i,j)\\&\quad-(t^2/2!-t^4/4!+t^6/6!+\cdots)\Lambda(i,j)\\&=\hat{I}_{ij}+\sin(t)\Omega(i,j)+\cos(t)\Lambda(i,j),\end{aligned}$$

where

$$\hat{1}_{ij}=I_n-\Lambda(i,j).$$

Moreover,

$$\Omega(i,j)\circ\hat{1}_{ij}=\Lambda(i,j)\circ\hat{1}_{ij}=\Omega(i,j)\circ\Lambda(i,j)=0. \tag{20}$$

Hence,

$$\begin{aligned} e^{t\Omega(i,j)}\circ e^{t\Omega(i,j)}&=\hat{1}_{ij}\circ\hat{1}_{ij}+\sin^2(t)\Omega(i,j)\circ\Omega(i,j)+\cos^2(t)\Lambda(i,j)\circ\Lambda(i,j)\\ &=I_n+\sin^2(t)\Pi(i,j) \end{aligned}$$

and

$$a_{ij}(t,\Theta)=\mathrm{tr}\left(\sum_{k=1}^{r}X_k^T(I_n+\sin^2(t)\Pi(i,j))\tilde{Y}_k(\Theta)(I_n+\sin^2(t)\Pi(i,j))\right), \tag{21}$$

if Θ is hyperoctahedral. □

Lemma 5.2. The set of critical points of Eq. (15) contains all the hyperoctahedral matrices.

Proof. It follows from (21) that, at a hyperoctahedral Θ,

$$\begin{aligned} da_{ij}(t,\Theta)/dt=2\sin(t)\cos(t)\,\mathrm{tr}\Bigg(\sum_{k=1}^{r}&(X_k^T\Pi(i,j)\tilde{Y}_k(\Theta)(I_n+\sin^2(t)\Pi(i,j))\\ &+X_k^T(I_n+\sin^2(t)\Pi(i,j))\tilde{Y}_k(\Theta)\Pi(i,j))\Bigg). \end{aligned} \tag{22}$$

Hence, for any $1\le i<j\le n$,

$$da_{ij}/dt(t,\Theta)|_{t=0}=0. \tag{23}$$

So, Θ is a critical point of η. □

Lemma 5.3. Under the basis defined by the matrices $\Omega(i,j)$, $H_\eta(\Theta)$ is a diagonal matrix if Θ is hyperoctahedral and the diagonal entries are of the form

$$H_{i,j}(\Theta)=2\,\mathrm{tr}\left(\sum_{k=1}^{r}(X_k^T\Pi_{i,j}\tilde{Y}_k(\Theta)+X_k^T\tilde{Y}_k(\Theta)\Pi_{i,j})\right), \tag{24}$$

for $1\le i<j\le n$.

Proof. Fix two distinct basis elements $\Omega(i,j)$ and $\Omega(l,m)$. Define

$$\tilde{\Omega}(s,t)=s\Omega(i,j)+t\Omega(l,m).$$

To show that the Hessian is diagonal, it is sufficient to show that the function

$$b(s,t)=\eta(\Theta\, e^{\tilde{\Omega}(s,t)})$$

has the property

$$\partial^2 b/\partial s\,\partial t(s,t)|_{s=t=0}=0. \tag{25}$$

By Lemma 5.1, $b(s,t)$ can be rewritten as

$$b(s,t)=\mathrm{tr}\left(\sum_{k=1}^{r} X_k^T(e^{\tilde{\Omega}(s,t)}\circ e^{\tilde{\Omega}(s,t)})^T\tilde{Y}_k(\Theta)(e^{\tilde{\Omega}(s,t)}\circ e^{\tilde{\Omega}(s,t)})\right). \tag{26}$$

Since, $b(s,t)$ is an analytic function of s and t, the property expressed by Eq. (25) is equivalent to the fact that the term st has coefficient zero in the power series expansion of $b(s,t)$. To prove the latter statement, we have to consider two separate cases.

If i, j, l, m are all distinct, then $\Omega(i,j)$ and $\Omega(l,m)$ are commuting matrices. Hence,

$$e^{\tilde{\Omega}(s,t)}=e^{s\Omega(i,j)}\,e^{t\Omega(l,m)} \tag{27}$$

and

$$\begin{aligned}
&e^{\tilde{\Omega}(s,t)}\circ e^{\tilde{\Omega}(s,t)}\\
&=(e^{s\Omega\,(i,j)}e^{t\Omega\,(l,m)})\circ(e^{s\Omega\,(i,j)}e^{t\Omega\,(l,m)})\\
&=((I_n+\sin^2(s)\Pi_{ij})(I_n+\sin^2(t)\Pi_{lm}))\\
&\quad\circ((I_n+\sin^2(s)\Pi_{ij})(I_n+\sin^2(t)\Pi_{lm}))\\
&=(I_n+\sin^2(s)\Pi_{ij}+\sin^2(t)\Pi_{lm})\circ(I_n+\sin^2(s)\Pi_{ij}+\sin^2(t)\Pi_{lm})\\
&=I_n+\sin^4(s)\Pi_{ij}\circ\Pi_{ij}+\sin^4(t)\Pi_{lm}\circ\Pi_{lm}.
\end{aligned} \tag{28}$$

If i, j, l, m are not distinct, then we may assume without loss of generality that $l=j$ and that i, j, m are distinct. In fact, we may assume that the two distinct basis elements are $\Omega(1,2)$ and $\Omega(2,3)$. Hence,

$$\tilde{\Omega}(s,t)=\begin{bmatrix} 0 & s & 0 & 0 & \dots & 0 \\ -s & 0 & t & 0 & \cdots & 0 \\ 0 & -t & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$$\tilde{\Omega}^2(s,t)=\begin{bmatrix} -s^2 & 0 & st & 0 & \cdots & 0 \\ 0 & -(s^2+t^2) & 0 & 0 & \cdots & 0 \\ st & 0 & -t^2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$$\tilde{\Omega}^3(s,t)=-(s^2+t^2)\tilde{\Omega}(s,t).$$

So,

$$e^{\tilde{\Omega}(s,t)}=I_n+u(s,t)\Omega(s,t)+v(s,t)\Omega^2(s,t), \tag{29}$$

where

$$u(s,t)=1-(s^2+t^2)/3!+(s^2+t^2)^2/5!-\cdots \tag{30}$$

$$v(s,t)=(1/2)-(s^2+t^2)/4!+(s^2+t^2)^2/6!-\cdots. \tag{31}$$

So,

$$e^{\tilde{\Omega}(s,t)}\circ e^{\tilde{\Omega}(s,t)}$$

$$=I_n+u^2(s,t)\begin{bmatrix} 0 & s^2 & 0 & 0 & \cdots & 0 \\ s^2 & 0 & t^2 & 0 & \cdots & 0 \\ 0 & t^2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$+v^2(s,t)\begin{bmatrix} s^4 & 0 & s^2t^2 & 0 & \cdots & 0 \\ 0 & (s^2+t^2)^2 & 0 & 0 & \cdots & 0 \\ s^2t^2 & 0 & t^4 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$-2v(s,t)\begin{bmatrix} s^2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & (s^2+t^2) & 0 & 0 & \cdots & 0 \\ 0 & 0 & t^2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{32}$$

For both cases, it is clear that all the entries of the matrix $e^{\tilde{\Omega}(s,t)} \circ e^{\tilde{\Omega}(s,t)}$ do not have the terms s, t, st in their power series expansion. It follows from Eq. (26) that the coefficient of the stth term in the power series expansion of $b(s,t)$ is zero. Finally, since the diagonal entries of $H_\eta(\Theta)$ are given by

$$d^2a_{i,j}(t,\Theta)/dt^2|_{t=0}=0, \tag{33}$$

it follows from Eq. (22) that they are of the form

$$2\,\mathrm{tr}\left(\sum_{k=1}^{r}(X_k^T\Pi_{i,j}\tilde{Y}_k(\Theta)+X_k^T\tilde{Y}_k(\Theta)\Pi_{i,j})\right). \qquad \square$$

We are now in a position to prove Theorem 5.1.

Proof of Theorem 5.1. First of all, observe that

$$\begin{aligned}\eta(\theta N_{i,j}) &= \mathrm{tr}\left(\sum_{k=1}^{r} X_k^T(I_n+\Pi_{ij})\tilde{Y}_k(\Theta)(I_n+\Pi_{ij})\right)\\ &= \eta(\Theta)+\mathrm{tr}\left(\sum_{k=1}^{r}(X_k^T\Pi_{ij}\tilde{Y}_k(\Theta)+X_k^T\tilde{Y}_k(\Theta)\Pi_{ij})\right)\\ &\quad+\mathrm{tr}\left(\sum_{k=1}^{r}X_k^T\Pi_{ij}\tilde{Y}_k(\Theta)\Pi_{ij}\right).\end{aligned} \tag{34}$$

Hence

$$\eta(\theta N_{ij}) - \eta(\theta) = (1/2)H_{ij} + \mathrm{tr}\left(\sum_{k=1}^{r} X_k^T \Pi_{ij} \tilde{Y}_k(\Theta)\Pi_{ij}\right). \tag{35}$$

Now, if P is a nondegenerate 2-opt solution, $\Theta_0 = \iota(P)$ is a hyperoctahedral matrix; hence, it is a critical point of η. By the assumption of the theorem,

$$H_{ij}(\Theta_0) > 0, \qquad \text{for all } 1 \leq i < j \leq n.$$

So, Θ is a nondegenerate local minimum of η on $\mathscr{SO}(n)$. Now, regard

$$V(\Theta) = \eta(\Theta) - \eta(\Theta_0)$$

as a Liapunov function on $\mathscr{SO}(n)$. It follows that, in a small enough neighborhood around Θ_0, $V > 0$ except at Θ_0. Moreover, from Eq. (15), it follows that $dV/dt \leq 0$. Hence, using the standard Liapunov stability theorem (Ref. 20), modified to the $\mathscr{SO}(n)$ context, if follows that Θ_0 is an asymptotically stable critical point of the gradient flow.

Conversely, assume that Θ_0 is a hyperoctahedral, asymptotical stable critical point. If the Hessian of η at Θ_0 is not nonnegative definite, then by the lemma of Morse (Ref. 21), in every small enough $\mathscr{SO}(n)$ neighborhood of Θ_0, there is a point Ψ such that $V(\Psi) < V(\Theta_0)$. Since $dV/dt \leq 0$, V is nonincreasing on any trajectory. So, the trajectory starting from Ψ cannot converge to Θ_0, a contradiction. Hence,

$$H_{ij}(\Theta_0) \geq 0 \text{ and } \eta(\Theta_0 N_{ij}) - \eta(\Theta_0) \geq 0, \qquad \text{for all } 1 \leq i < j \leq n. \qquad \square$$

At first glance, the condition in Theorem 5.1 may be very difficult to verify. However, observe that, if

$$\mathrm{tr}\left(\sum_{k=1}^{r} X_k^T \Pi_{i,j} \tilde{Y}_k(\Theta)\Pi_{i,j}\right) = 0,$$

then the conditions (16) hold automatically. For any $n \times n$ matrix M, define

$$\Delta_{ij}(M) = M_{ij} + M_{ji} - M_{ii} - M_{jj}. \tag{36}$$

Then, it is straightforward to check that

$$\mathrm{tr}(X_k^T \Pi(i,j)\tilde{Y}_k(\Theta)\Pi(i,j) = \Delta_{i,j}(X_k)\Delta_{i,j}(\tilde{Y}_k(\Theta))). \tag{37}$$

Moreover, the following result holds.

Theorem 5.2. For any $1 \leq k \leq r$, if $\Delta_{ij}(X_k) = 0$ for all i, j, or if $\Delta_{ij}(Y_k) = 0$ for all i, j, then the weak gradient equivalence property holds for the problem defined by (10).

Proof. If Θ is a hyperoctahedral matrix, define ϕ to be the one-to-one mapping, from the set $\{1, \dots, n\}$ to itself, by the implicit requirement that $\Theta_{\phi(i)i} \neq 0$. Since Θ is hyperoctahedral, ϕ is well defined and

$$\begin{aligned}\Delta_{ij}(\tilde{Y}_k(\Theta)) &= (\tilde{Y}_k(\Theta))_{ij} + (\tilde{Y}_k(\Theta))_{ji} - (\tilde{Y}_k(\Theta))_{ii} - (\tilde{Y}_k(\Theta))_{jj} \\ &= (Y_k)_{\phi(i)\phi(j)} + (Y_k)_{\phi(j)\phi(i)} - (Y_k)_{\phi(i)\phi(i)} - (Y_k)_{\phi(j)\phi(j)} \\ &= \Delta_{\phi(i)\phi(j)}(\tilde{Y}_k(\Theta)). \end{aligned} \tag{38}$$

Hence, if either of the stated conditions hold, then

$$\operatorname{tr}\left(\sum_{k=1}^{r} X_k^T \Pi(i,j)^T \tilde{Y}_k(\Theta) \Pi(i,j)\right) = 0,$$

and the result follows from Theorem 5.1. □

Define a matrix M to be locally balanced if $\Delta_{ij}(M) = 0$ for all i, j. It is easy to show that a matrix is locally balanced if and only if it is of the form

$$M = A + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} c^T, \tag{39}$$

where A is antisymmetric and c is an arbitrary n-dimensional row vector.

In the next section, we will present specific examples where the conditions of Theorem 5.2 are satisfied.

6. Weak Gradient Equivalence Property of the Graph Partitioning Problem

We have discussed previously how to represent the graph partitioning problem as a problem on $\mathscr{P}_n$. If we define $S_{\mathscr{G}}(n)$ in a slightly different way, by adding the identity matrix to the right-hand side of Eq. (8), the new cost function $\operatorname{tr}(C^T P^T (I_n + S_{\mathscr{G}}(n)) P)$ differs from the old function by a fixed constant, $\operatorname{tr}(C)$. The new optimization problem defined on this function is clearly equivalent to the original problem. For reasons to be explained later, we will use this definition of $S_{\mathscr{G}}$ to represent a graph partitioning problem. Hence, from now on, we define

$$S_{\mathscr{G}}(n) = \begin{bmatrix} I_p & 1_{pq} \\ 1_{qp} & I_q \end{bmatrix}. \tag{40}$$

We may assume also without loss of generality that all the diagonal elements of C are zero and all entries of C are nonnegative.

Theorem 6.1. All graph partitioning problems, with the cost matrix modified to have nonnegative entries and zero diagonal elements, have the weak gradient equivalence property.

Proof. It is easy to check that, for all $1 \le i < j \le n$,

$$\Delta_{ij}(C) \ge 0 \quad \text{and} \quad \Delta_{ij}(\tilde{S}_{\mathscr{G}}(\Theta)) = 0 \text{ or } -2,$$

if Θ is hyperoctahedral. If $\Delta_{ij}(\tilde{S}_{\mathscr{G}}(\Theta)) = 0$, then

$$\operatorname{tr}(C^T \Pi(i,j) \tilde{S}_{\mathscr{G}}(\Theta) \Pi(i,j)) = 0,$$

and the two conditions stated in (16) hold. If $\Delta_{ij}(\tilde{S}_{\mathscr{G}}(\Theta)) = -2$, then

$$\operatorname{tr}(C^T \Pi(i,j) \tilde{S}_{\mathscr{G}}(\Theta) \Pi(i,j)) < 0,$$

and the first condition of (16) clearly holds. Moreover, $\Delta_{ij}(\tilde{S}_{\mathscr{G}}(\Theta)) = -2$ also implies that

$$((\Theta \circ \Theta)^T S_{\mathscr{G}}(\Theta \circ \Theta))_{ij} + ((\Theta \circ \Theta)^T S_{\mathscr{G}}(\Theta \circ \Theta))_{ji} = 0. \tag{41}$$

Hence, each of the two terms on the left-hand side of Eq. (41) must vanish, and

$$((\Theta N_{ij}) \circ (\Theta N_{ij}))^T S_{\mathscr{G}}((\Theta N_{ij}) \circ (\Theta N_{ij})) = (\Theta \circ \Theta)^T S_{\mathscr{G}}(\Theta \circ \Theta). \tag{42}$$

So,

$$\eta(\Theta N_{ij}) = \eta(\Theta),$$

and the second condition stated in (16) also holds. $\square$

Notice that, although $\operatorname{tr}(C^T P^T (I_n + M) P)$ differs from $\operatorname{tr}(C^T P^T M P)$ by a fixed constant, this is not true for the embedded functions $\operatorname{tr}(C^T (\Theta \circ \Theta)^T (I_n + M)(\Theta \circ \Theta))$ and $\operatorname{tr}(C^T (\Theta \circ \Theta)^T M (\Theta \circ \Theta))$. Thus, choosing the new $S_{\mathscr{G}}$ is important. We do not expect that this argument for the weak equivalence property can be extended to the generalized graph partitioning problem.

7. Flows on One-Parameter Subgroups

If we restrict the function η to a one-parameter subgroup $\Theta\, e^{t\Omega(i,j)}$, where Θ is a hyperoctahedral matrix, and define the function to be a_{ij} as in Eq. (18), it is obvious that a_{ij} has a period of π. Moreover, if

$$\operatorname{tr}\left(\sum_{k=1}^{r} (X_k^T \Pi(i,j) \tilde{Y}_k(\Theta) \Pi(i,j)) \right) = 0,$$

then

$$da_{ij}(t, \Theta)/dt$$
$$= 2 \sin t(t) \cos (t) \operatorname{tr}\left(\sum_{k=1}^{r} (X_k^T \Pi(i,j) \tilde{Y}_k(\Theta) + X_k^T \tilde{Y}_k(\Theta) \Pi(i,j)) \right).$$

Hence, its critical points are located at 0, $\pi/2$, $3\pi/2$, and so on. It follows that, if we project the gradient vector field of η onto Ω_{ij} and restrict the starting point to be on the one-parameter subgroup, then the only critical points are Θ and ΘN_{ij}. Thus, one can mimic the local search procedure by projecting cyclically the gradient vector field onto the Ω_{ij} and following the flows on the corresponding one-parameter subgroups.

8. Numerical Studies of TSP

The condition of Theorem 5.2 does not hold for a traveling salesman problem unless the cost matrix is locally balanced. The investigate the effectiveness and efficiency of the gradient flow approach for solving the Euclidean traveling salesman problem, Wu performed some numerical studies in Ref 22. In order to test how restrictive the locally balanced condition is, general symmetric cost matrices were used. His study shows that, for problems with 10 to 20 cities, a locally optimal tour can be obtained routinely quite efficiently, even though the weak gradient equivalence property may not hold. For problems with 30 cities, nonpermutation local solutions may be obtained. It is relatively easy to obtain permutation matrix approximations from these nonpermutation local solutions, and these approximations turn out to be nearly locally optimal. While our analysis for the traveling salesman problem is limited (providing only a weak equivalence result for cost matrices that are locally balanced, our approach leads nevertheless to an analog, easily parallelizable alternative for solving some small-sized problems.

References

1. Bayer, D. A., and Lagarias, J. C., *The Nonlinear Geometry of Linear Programming, Parts 1 and 2*, Transactions of American Mathematical Society, Vol. 314, pp. 499–526, 1989 and Vol. 314, pp. 527–581, 1989.
2. Bloch, A. M., *Steepest Descent, Linear Programming, and Hamiltonian flows*, Mathematical Developments Arising from Linear Programming, Edited by J. C. Lagarias and M. J. Todd, Contemporary Mathematics, Vol. 114, pp. 297–308, 1990.

3. BROCKETT, R. W., *Dynamical Systems That Sort and Solve Linear Programming Problems*, Linear Algebra and Its Applications, Vol. 146, pp. 79–91, 1991.
4. BROCKETT, R. W., *Least Squares Matching Problems*, Linear Algebra and Its Applications, Vol. 122, pp. 761–777, 1989.
5. BROCKETT, R. W., and WONG, W. S., *A Gradient Flow for the Assignment Problem*, New Trends in Systems Theory, Edited by G. Conte and B. Wyman, Birkhäuser, Boston, pp. 170–177, 1991.
6. CHU, M. T., *On the Continuous Realization of Iterative Processes*, SIAM Review, Vol. 30, pp. 375–389, 1988.
7. DEIFT, T., NANDA, T., and TOMEI, C., *Differential Equations for the Symmetric Eigenvalue Problem*, SIAM Journal on Numerical Analysis, Vol. 20, pp. 1–22, 1983.
8. FAYBUSOVICH, L., *Hamiltonian Structure of Dynamical Systems Which Solve Linear Programming Problems*, Physica D, Vol. 53, pp. 217–232, 1991.
9. FAYBUSOVICH, L., *Dynamical Systems Which Solve Optimization Problems with Linear Constraints*, IMA Journal of Mathematical Control and Information, Vol. 8, pp. 135–149, 1991.
10. SYMES, W. W., *Hamiltonian Group Actions and Integrable Systems*, Physica D, Vol. 1, pp. 339–376, 1980.
11. SYMES, W. W., *The QR Algorithm and Scattering for the Nonperiodic Toda Lattice*, Physica D, Vol. 4, pp. 275–280, 1982.
12. WATKINS, D. S., and ELSNER, L., *On Rutishauser's Approach to Self-Similar Flows*, SIAM Journal on Matrix Analysis and Applications, Vol. 11, pp. 301–311, 1990.
13. KARMARKAR, N., *An Interior-Point Approach to NP-Complete Problems, Part 1*, Mathematical Developments Arising from Linear Programming, Edited by J. C. Lagarias and M. J. Todd, Contemporary Mathematics, Vol. 114, pp. 297–308, 1990.
14. KAMATH, A. P., KARMARKAR, N. K., RAMAKRISHNAN, K., and RESENDE, M. G. C., *Computational Experience with an Interior-Point Algorithm on the Satisfiability Problems*, Annals of Operations Research, Vol. 25, pp. 43–58, 1990.
15. KARMARKAR, N. K., RAMAKRISHNAN, K., and RESENDE, M. G. C., *An Interior-Point Algorithm to Solve Computationally Difficult Set-Covering Problems*, Mathematical Programming, Vol. 52B, pp. 597–618, 1991.
16. KARMARKAR, N. K., and THAKUR, S. A., *An Interior-Point Approach to Tensor Optimization Problems with Application to Upper Bounds in Integer Quadratic Optimization Problems*, Proceedings of the 2nd Conference on Integer Programming and Combinatorial Optimization, pp. 406–420, 1992.
17. SPIVEY, W. A., and THRALL, R. M., *Linear Optimization*, Holt, Rinehart, and Winston, New York, New York, 1970.
18. JOHNSON, D. S., PAPADIMITRIOU, C. H., and YANNAKAKIS, M., *How Easy Is Local Search?*, Journal of Computer and System Sciences, Vol. 37, pp. 79–100, 1988.
19. SCHÄFFER, A., and YANNAKAKIS, M., *Simple Local Search Problems That Are Hard to Solve*, SIAM Journal on Computing, Vol. 20, pp. 56–87, 1991.

20. Milnor, J., *Morse Theory*, Princeton University Press, Princeton, New Jersey, 1969.
21. Davies, T. V., and James, E. M., *Nonlinear Differential Equations*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1966.
22. Wu, N. L., *Analog Combinatorial Optimization*, Final Year Student Report, Chinese University of Hong Kong, 1992.