LibCoopt: A library for combinatorial optimization on partial permutation matrices

Zhi-Yong Liua,c,d,⁎, Jing-Jing Wanga, Xu Yanga, Shi-Hao Fenge

a State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
b School of Electronic and Information Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
c CAS Centre for Excellence in Brain Science and Intelligence Technology (CEBSIT), Shanghai 200031, China
d School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO
Communicated by Zhu Jianke
Keywords:
Combinatorial optimization
Partial permutation matrices
Graduated projection
Deterministic annealing

ABSTRACT
LibCoopt is an open-source matlab code library which provides a general and convenient tool to approximately solve the combinatorial optimization problems on the set of partial permutation matrices, which are frequently encountered in computer vision, bioinformatics, social analysis, etc. To use the library, the user needs only to give the objective function and its gradient function associated with the problem. Two typical problems, the subgraph matching problem and the quadratic assignment problem, are employed to illustrate how to use the library and also its flexibility on different types of problems.

1. Introduction
Combinatorial optimization on partial permutation matrices plays a key role in many computer science problems, such as the subgraph matching problem (SGM) and the quadratic assignment problem (QAP). These problems are usually NP-hard, and therefore some approximations are necessary for efficiency reasons [1,2]. In literature, these problems were usually solved by different specifically designed methods. In this paper we try to handle these problems from a unified viewpoint. Specifically, the graduated nonconvexity and concavity procedure (GNCCP) [3], proposed by us previously, is adopted as the combinatorial optimization algorithmic framework. An important advantage of GNCCP is that only the objective function and its gradient function are involved when it is applied to combinatorial optimization problems. Based on GNCCP, in this paper we introduce an open-source matlab code library, LibCoopt, which provides a general and convenient tool for combinatorial optimization on partial permutation matrices. The software is applied to two typical combinatorial optimization problems, SGM and QAP, to show how to use it, as well as to illustrate its flexibility.

⁎ Corresponding author.
E-mail address: zhiyong.liu@ia.ac.cn (Z.-Y. Liu).

https://dx.doi.org/10.1016/j.neucom.2016.11.048
Received 27 June 2016; Received in revised form 14 November 2016; Accepted 24 November 2016
Available online 08 December 2016
0925-2312/ © 2016 Elsevier B.V. All rights reserved.
2. Problems and background

2.1. Formulation

We consider the combinatorial optimization problem defined by
\[ \min F(X), \text{ s.t. } X \in \Omega, \Omega = \{XX = 0, 1\}, \sum_{j=1}^{N} X_{j} = 1, \sum_{i=1}^{M} X_{i} \leq 1, \forall i, j \}, \quad M \leq N, \quad (1) \]

where \( \Omega \) is the set of partial permutation matrices of the size \((M \times N)\). The objective function \( F(X) \) is assumed to be differentiable. Such a formulation covers a wide range of important problems, such as SGM and QAP.

2.2. Background and related works

The LibCoopt has its root in GNCCP. The GNCCP generalizes the convex-concave relaxation procedure (CCRP) [1,4], which involves a linear combination of a convex relaxation and a concave relaxation of the original objective function, and exhibited superior performance on the equal-sized graph matching problem. However, it is not trivial to generalize the CCRP to other combinatorial optimization problems because of the difficulties in finding the optimal solution. It was shown that the GNCCP equivalently realizes a type of CCRP on partial permutation matrices, but in a much simpler way without explicitly involving the convex or concave relaxation.

To use GNCCP, \( \Omega \) is firstly relaxed to its convex hull \( \Omega_{c} = \{XX \geq 0, \sum_{j=1}^{N} X_{j} = 1, \sum_{i=1}^{M} X_{i} \leq 1, \forall i, j \} \). Then the GNCCP takes the following form
\[ F_{c}(X) = \begin{cases} (1 - \zeta)F(X) + \zeta uu^T & \text{if } 1 \leq \zeta \geq 0, \\ (1 + \zeta)F(X) + \zeta uu^T & \text{if } 0 > \zeta \geq -1, \end{cases} \quad X \in \Omega. \quad (2) \]

The algorithmic framework for GNCCP is given by Algorithm 1. In the algorithm, the gradient \( \nabla F_{c}(X) \) takes the following form
\[ \nabla F_{c}(X) = \begin{cases} (1 - \zeta)\nabla F(X) + 2\zeta uX & \text{if } 1 \geq \zeta \geq 0, \\ (1 + \zeta)\nabla F(X) + 2\zeta uX & \text{if } 0 > \zeta \geq -1. \end{cases} \quad (3) \]

Algorithm 1. Algorithmic framework of GNCCP.

\[ \begin{align*} & \zeta \leftarrow 1, X \leftarrow X^0 \; \text{while } \zeta > -1 \text{ and } X \in \Omega \Rightarrow \; \text{while } X \text{ not converged} \Rightarrow \; \\
& \quad \Rightarrow Y = \arg \min_{Y} \nabla F_{c}(X)^{T}Y, \text{ s.t. } Y \in \Omega. \\
& \quad \Rightarrow \alpha = \arg \min_{\alpha} F_{c}(X + \alpha(Y - X)), \text{ s.t. } \alpha \in [0, 1] \\
& \quad \Rightarrow x \leftarrow X + \alpha(Y - X) \; \text{end while } \Rightarrow \; \\
& \quad \Rightarrow \zeta \leftarrow \zeta - \alpha \zeta \; \text{end while } \Rightarrow \end{align*} \]

As shown in Algorithm 1, the GNCCP involves only the objective function \( F_{c}(X) \) and its gradient \( \nabla F_{c}(X) \). Therefore it provides a unified framework for quite a lot of combinatorial optimization problems on partial permutation matrices as long as the objective function is differentiable. Based on GNCCP, the LibCoopt is introduced below.

3. Software architecture and implementation

The architecture of LibCoopt is shown in Fig. 1. For different combinatorial optimization problems on partial permutation matrices, LibCoopt provides an interface to input both the objective function and its gradient function. The two functions take the problem related data as input, while LibCoopt itself does not directly face the data. It is in this sense that LibCoopt is claimed to be a general and convenient tool for combinatorial optimization on partial permutation matrices.

LibCoopt is mainly implemented by Matlab script, with some computationally intensive parts implemented by Mex files. Currently only the Windows operation system based version is provided.

The core Matlab function is
\[ \text{Solution} = \text{Coopt}(@F, \text{@nF}, \text{Data, Para}) \]

where Solution is the final combinatorial optimization solution including the minimal point, objective value, and running time. The first two inputs \( @F \) and \( @nF \) are the function handles of the customized objective function and its gradient function. The third input Data is the problem related data. And Para is the parameter structure.

To show how to use LibCoopt in a specific problem, the objective functions and their gradient functions of two typical combinatorial optimization problems, i.e. SGM and QAP, are provided in the library. Before introducing the corresponding Matlab functions, some brief preliminaries are given below. For the adjacency matrix based SGM model (denoted by GMAD) [3], the objective function is
\[ \text{GMAD: } \min F(X) = \|A_{m} - XA_{T}X^{T}\|_{F} \; \text{s.t. } X \in \Omega, \quad (4) \]

where \( A_{m} \) and \( A_{T} \) denote the adjacency matrices associated with the two input graphs. For the affinity matrix based SGM model (denoted by GMAF) [5], the objective function is
\[ \text{GMAF: } \min F(X) = vec(X)^{T}A_{m}vec(X) \text{ or } F(X) = vec(X)^{T}Kvec(X) \; \text{s.t. } X \in \Omega, \quad (5) \]

where \( A \) is a \( MN \times MN \) affinity matrix encoding the edge similarities between graphs, and similarly \( K \) denotes the dissimilarity matrix which can be directly obtained by \( K = -A \). We use the minimization problem based on \( K \) in this paper. For QAP [3], the objective function is
\[ \text{QAP: } F(X) = tr(AXB^{T}X^{T}) \; \text{s.t. } X \in \Omega(M = N) \quad (6) \]

where \( A \) and \( B \) are two equal-sized matrices.

In LibCoopt, for the objective function and gradient function of GMAD, the corresponding Matlab functions are \( F_{-}\text{GMAD}(X,\text{Data}) \) and \( nF_{-}\text{GMAD}(\text{Data}) \). For GMAF, they are \( F_{-}\text{GMAF}(X,\text{Data}) \) and \( nF_{-}\text{GMAF}(\text{Data}) \), and for QAP, they are \( F_{-}\text{QAP}(X,\text{Data}) \) and \( nF_{-}\text{QAP}(\text{Data}) \).

The users may directly try run_Coopt_GMAD(DataPath), run_Coopt_GMAF(DataPath), and run_Coopt_QAP(DataPath) to test LibCoopt on these problems, where DataPath is the path of a sample data. These functions call Coopt in similar ways, and provide data preprocessing and other specific processing for different problems. Users can use data in folder ToyData for testing. More demos and description can be found at https://github.com/RowenaWong/libcoopt.

Moreover, by specifying the files in the folder Other, LibCoopt can be also applied to other combinatorial optimization problem as long as it can be formulated into a differentiable objective function on partial permutation matrices. Taking the traveling salesman problem (TSP) for example, it can be formulated by
\[ F(X) = \|FDX^{T}X^{T}\|_{F} \; \text{s.t. } X \in \Omega \quad (M = N), \quad (7) \]

where \( D \) is the distance matrix and \( F \) is a constant matrix defined by
\[ F_{ij} = \begin{cases} 1 & \text{if } j = i + 1 \text{ or } i = N, j = 1, \\
0 & \text{otherwise}. \end{cases} \quad (8) \]

This formulation is similar to QAP and the derivation of its gradient is straightforward. Thus by formulating TSP in this way, LibCoopt is applicable to it.
4. Empirical results

LibCoopt is evaluated by applying it to GMAD, GMAF, and QAP on synthetic graphs and real-world data. The users can run exp_GMAD, exp_GMAF, exp_QAP('sym'), and exp_QAP('asym') to repeat these experiments.

4.1. GMAD

Graph pairs are generated by the Matlab function SData. LibCoopt is evaluated with respect to noise level which is increased from 0 to 0.2 by a step size of 0.02. For each noise level, 10 pairs of graphs are generated by SData with 20 inliers and 5 outliers. The parameter setting is as follows: the learning step $\eta = 0.002$, the stopping parameter $\eta = 0.001$. The criterion is the matching accuracy (acc%). The results are shown in Table 1.

4.2. GMAF

In this experiment LibCoopt is applied to GMAF on four handwritten Chinese characters [6]. Some matching instances are shown in Fig. 2.

4.3. QAP

LibCoopt is evaluated on the symmetric and asymmetric QAPLIB benchmark datasets [7]. The parameter settings are as follows: $d_\zeta = 0.001$, $\eta = 0.001$. The criterion is average wrong assignment ratio, $awar(\%) = \frac{1}{n} \sum_{i=1}^{n} \frac{cost_i - opt_i}{opt_i}$. The experimental results are showed in Table 2.

5. Illustrative examples

Some other illustrative examples of LibCoopt can be found at: https://github.com/RowenaWong/libcoopt.

Table 1

<table>
<thead>
<tr>
<th>Noise level</th>
<th>0</th>
<th>0.02</th>
<th>0.04</th>
<th>0.06</th>
<th>0.08</th>
<th>0.10</th>
<th>0.12</th>
<th>0.14</th>
<th>0.16</th>
<th>0.18</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>acc(%)</td>
<td>97.5</td>
<td>99.0</td>
<td>97.0</td>
<td>95.5</td>
<td>96.5</td>
<td>95.0</td>
<td>93.5</td>
<td>94.0</td>
<td>94.5</td>
<td>94.0</td>
<td>92.5</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Data (sym)</th>
<th>chr12b</th>
<th>chr15c</th>
<th>chr20b</th>
<th>chr25a</th>
<th>rou15</th>
<th>rou20</th>
<th>tail15b</th>
<th>tail17a</th>
<th>awar (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPT</td>
<td>9742</td>
<td>9504</td>
<td>2298</td>
<td>3796</td>
<td>354210</td>
<td>725522</td>
<td>5176268</td>
<td>491812</td>
<td>14.9</td>
</tr>
<tr>
<td>LibCoopt</td>
<td>11636</td>
<td>12520</td>
<td>2654</td>
<td>5292</td>
<td>364680</td>
<td>747944</td>
<td>5275500</td>
<td>517618</td>
<td>14.9</td>
</tr>
<tr>
<td>Data (asym)</td>
<td>lipa20a</td>
<td>lipa20b</td>
<td>lipa30a</td>
<td>lipa30b</td>
<td>lipa40a</td>
<td>lipa40b</td>
<td>lipa50a</td>
<td>lipa50b</td>
<td>awar (%)</td>
</tr>
<tr>
<td>OPT</td>
<td>3683</td>
<td>27076</td>
<td>13178</td>
<td>151426</td>
<td>31538</td>
<td>476581</td>
<td>62847</td>
<td>1210244</td>
<td>1.0</td>
</tr>
<tr>
<td>LibCoopt</td>
<td>3802</td>
<td>27076</td>
<td>13437</td>
<td>151426</td>
<td>32027</td>
<td>476581</td>
<td>62847</td>
<td>1210244</td>
<td>1.0</td>
</tr>
</tbody>
</table>
6. Conclusions

We introduce an open-source matlab code library, LibCoopt, which provides a general and convenient tool for combinatorial optimization on partial permutation matrices. Two typical problems, SGM and QAP, are employed to show how to use LibCoopt in detail. In the future we are to add more cases of combinatorial optimization to LibCoopt and keep improving its performance.

Acknowledgements

The work described in this paper is supported by the National Foundation of Science of China (NFSC) (grants 61375005, U1613213, 61033011), and Ministry of Science and Technology of China (MOST) (grants 2015BAK35B00, 2015BAK35B01)

References


Zhi-Yong Liu is a professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include machine learning, pattern recognition, computer vision, and bioinformatics.

Jing-Jing Wang received her Bachelor’s degree from Nankai University, China, in 2014. She is a master candidate in the Institute of Automation, Chinese Academy of Sciences (IA-CAS), China. Her current research interests include online learning and graph algorithms.

Xu Yang is an assistant professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, graph algorithms, and robotics.

Shi-Hao Feng received his Bachelor’s degree from Xi’an Jiaotong University, China, in 2016. His current research interests include complex network and bioinformatics.