A comparative study of linearization methods for Ordered Weighted Average

Thi Quynh Trang Vo, Mourad Baiou, Viet Hung Nguyen, Paul Weng

A comparative study of linearization methods for Ordered Weighted Average

Thi Quynh Trang Vo*, Mourad Baiou*, Viet Hung Nguyen*‡, Paul Weng†
* INP Clermont Auvergne, Univ Clermont Auvergne, Mines Saint-Etienne,
CNRS, UMR 6158 LIMOS, Aubiere Cedex, France
† Shanghai Jiao Tong University, Shanghai, China
UM-SJTU Joint Institute, Shanghai, China
‡ Corresponding author
Email: thi_quynh_trang.vo@uca.fr, mourad.baiou@uca.fr, viet_hung.nguyen@uca.fr, paul.weng@sjtu.edu.cn

***Abstract*—We consider a fair version of combinatorial optimization, which aims for both Pareto-efficiency and fairness of a solution. A possible approach to achieve the objectives simultaneously is to use the Ordered Weighted Average (OWA) aggregating function, which can be formulated into mix-integer programming (MIP) formulations. In this paper, we study two MIP formulations proposed in the literature for the OWA in the context of fair combinatorial optimization. On the one hand, we prove that both MIP formulations are equivalent in terms of linear relaxations. On the other hand, we estimate the quality with regard to the OWA value of an optimal solution of original combinatorial optimization. An experimental evaluation of the MIP formulations in tackling OWA Traveling Salesman Problem is also presented.**

***Index Terms*—Fair Optimization, Ordered Weighted Averaging, Subgradient Method**

I. Introduction

Fair combinatorial optimization is a class of combinatorial optimization where one seeks a solution of the form $(v_1, \dots, v_n)$ satisfying some constraints to optimize an objective function $f(v) = \sum_{i\in[n]} v_i$ as well as to *balance* all the components $v_i, i = 1, \dots, n$ (in the sense that they are close to each other). It arises naturally in many practical applications. In fact, one can view fair combinatorial optimization as an equitable version of combinatorial optimization where the optimality of the objective function f and the balance of the solution $(v_1, \dots, v_n)$ are equally desired.

To illustrate one idea for fair combinatorial optimization, we take OWA Travelling Salesman Problem (TSP) as an example: given a weighted graph, OWA TSP focuses on simultaneously minimizing (respectively maximizing) the total cost (respectively the total quality) of a tour as well as making $v_i, i = 1, \dots, n$ balanced (v_i is the cost/quality of an edge in the tour). Such a scenario is realized, for example, in a telecommunications network where each node denotes a server, and each edge denotes a possible direct link between two servers with some quality represented by the weight of the corresponding edge. The aim is to find a ring (i.e., a Hamiltonian cycle) of direct links connecting the servers. Any pair of servers can communicate via one of the two paths connecting them in the ring, and the connection quality depends on the edge of minimum quality in the paths. Hence, the ring structure allows having a secured connection when one of the direct links fails. Moreover, one may want to guarantee some balance in the connection quality between all the pairs of servers. In this situation, the maximum version of OWA TSP can offer a good solution: it maximizes the total quality of the tour, and at the same time, it guarantees some balance in the quality of the connections between any pair of servers. For the sake of clarity, in this paper, we focus on minimization combinatorial optimization problems since the maximization counterparts can be handled similarly.

One approach for controlling solutions' efficiency and fairness is to use the Ordered Weighted Average (OWA) function [3]. The OWA function encodes the solution's efficiency by Pareto optimality that is not improvable on all components simultaneously. On the other hand, the OWA function implicitly imposes the solution's fairness by Pigou-Dalton transfer principle, which claims that a transfer from a richer resource to a poorer one results in a fairer distribution. We call *OWA-combinatorial optimization* an equitable version of combinatorial optimization, which uses the OWA objective function instead of the sum function.

Although the OWA function is non-linear, it can be linearized by two methods: one proposed by Ogryczak et al. [4] using the cumulative ordered achievement vector and the other given by Chassein et al. [1] employing the permutahedron. The results of these linearization methods are two MIP formulations for OWA-combinatorial optimization. From now on, let us denote O-MIP, the MIP resulted from the method in [4] and C-MIP, the MIP resulted from the method in [1]. In this paper, we consider C-MIP, the more recent formulation, as O-MIP has been studied in [3]. Furthermore, we provide a comparative study of the two MIP formulations for OWA-combinatorial optimization. In detail, while C-MIP has fewer variables and is reported to be more efficient than O-MIP for some problems like OWA simplified portfolio optimization [1], we prove that O-MIP and C-MIP are equivalent in terms of linear relaxations. In addition, for OWA TSP, O-MIP can be solved faster than C-MIP despite its disadvantage in the number of variables.

To summary, the contributions of this paper are:

1) We prove that the MIP formulations for OWA-

combinatorial optimization are equivalent in terms of linear relaxations.
2) We estimate the optimal solution's quality of original combinatorial optimization in terms of OWA value.
3) We experimentally compare O-MIP, C-MIP, and primal-dual heuristics [3] in the context of OWA TSP.

The paper is organized as follows. In Section II, we define formally OWA-combinatorial optimization and two MIP formulations to linearize it. Section III provides a theoretical analysis of the relation between the MIP formulations and a quality evaluation of the optimal solution of the original combinatorial optimization. Section IV presents a generic method based on the primal-dual algorithm in [3] as a faster alternative to deal with large-sized instances. Experimental results to evaluate the methods are shown in Section V. Finally, our conclusions are discussed in Section VI.

II. Models

Let us begin with some notations. For a positive integer n, $[n]$ stands for the set $\{1,\ldots,n\}$. Vectors and matrices are denoted in bold (e.g., $\boldsymbol{x}$ or $\boldsymbol{z}$) and their components are denoted with indices (e.g., x_i or z_{ij}).

A. OWA-combinatorial optimization

We consider a combinatorial optimization problem which has the following form:

$$(Min-\mathcal{P})\qquad \begin{aligned} &\min \sum_{i\in[n]} v_i && \text{(1a)}\\ &\text{s.t } \boldsymbol{v}=\boldsymbol{C}\boldsymbol{x} && \text{(1b)}\\ &\boldsymbol{A}\boldsymbol{x}\le\boldsymbol{b} && \text{(1c)}\\ &\boldsymbol{x}\in\{0,1\}^m && \text{(1d)}\end{aligned}$$

where $\boldsymbol{v}\in\mathbb{R}^n$ is the vector whose components are needed to be balanced, $\boldsymbol{C}\in\mathbb{R}^{n\times m}$, $\boldsymbol{A}\in\mathbb{R}^{K\times m}, \boldsymbol{b}\in\mathbb{R}^K$ and K,m,n are positive integers.

Note that $(Min-\mathcal{P})$ does not necessarily search for the balance of $\boldsymbol{v}$ since it only optimizes $\sum_{i\in[n]} v_i$. To obtain the balance, we introduce the notion of Ordered Weighted Average (OWA) value of $\boldsymbol{v}$, which is defined as:

$$\mathrm{OWA}_{\boldsymbol{w}}(\boldsymbol{v})=\sum_{k\in[n]} w_k\theta_k(\boldsymbol{v})$$

where $\boldsymbol{w}=(w_1,\ldots w_n)$ such as $w_1>\cdots>w_n>0$, $\theta(\boldsymbol{v})=(\theta_1(\boldsymbol{v}),\ldots,\theta_n(\boldsymbol{v}))$ is a vector obtained by arranging $\boldsymbol{v}$'s components in the descending order. By choosing weight $\boldsymbol{w}$ positive and strictly decreasing, the OWA function can encode both the efficiency and fairness of vector $\boldsymbol{v}$. The notion of efficiency is defined through the increase of the OWA function with respect to Pareto-dominance. In detail, if $\boldsymbol{y}\in\mathbb{R}^n$ Pareto-dominates $\boldsymbol{y}'\in\mathbb{R}^n$ ($y_i\ge y'_i\,\forall i\in[n]$, $\exists j\in[n], y_j>y'_j$) then $\mathrm{OWA}_{\boldsymbol{w}}(\boldsymbol{y})>\mathrm{OWA}_{\boldsymbol{w}}(\boldsymbol{y}')$. The OWA function can also represent the concept of fairness built on Pigou-Dalton principle. Formally, for $\boldsymbol{y}\in\mathbb{R}^n$ where $y_i>y_j$, for all $\epsilon\in(0,y_i-y_j)$, $\mathrm{OWA}_{\boldsymbol{w}}((y_1,...,y_i,...,y_j,...,y_n))>\mathrm{OWA}_{\boldsymbol{w}}((y_1,...,y_i-\epsilon,...,y_j+\epsilon,...,y_n))$.

An OWA-combinatorial optimization problem is defined as follows:

$$(Fair-\mathcal{P})\qquad \begin{aligned} &\min \mathrm{OWA}_{\boldsymbol{w}}(\boldsymbol{v}) && \text{(2a)}\\ &\text{s.t } \boldsymbol{v}=\boldsymbol{C}\boldsymbol{x} && \text{(2b)}\\ &\boldsymbol{A}\boldsymbol{x}\le\boldsymbol{b} && \text{(2c)}\\ &\boldsymbol{x}\in\{0,1\}^m && \text{(2d)}\end{aligned}$$

For illustration, we now present the concept of the OWA function in the context of TSP.

Example 1: Given a complete graph $G=(V,E)$ where $V=[n]$, $E=[m]$ and a cost vector $\boldsymbol{c}\in\mathbb{R}^m_+$ associated with E, TSP seeks n edges that form a Hamiltonian cycle (a tour) in G and have the smallest total cost. In other words, TSP minimizes $\sum_{i\in[n]} v_i$ where $\boldsymbol{v}=(v_1,\ldots,v_n)$ is the cost vector of n edges in a tour. To represent $\boldsymbol{v}$, we consider a directed version $G^d=(V,E^d)$ of G, namely that each edge $e=(i,j)\in G$ becomes two arcs (i,j) and (j,i) whose costs are both c_e, i.e., $c_{ij}=c_{ji}=c_e$. For each arc (i,j), let x_{ij} be a binary variable to represent the occurrence of (i,j) in a directed tour of G^d. By the fact that there is only one incoming arc incident to vertex k in a directed tour, i.e., $\sum_{i\in[n]} x_{ik}=1,\forall k\in[n]$, $\boldsymbol{v}$ can be represented as $(\sum_{i\in[n]} c_{i1}x_{i1},\ldots,\sum_{i\in[n]} c_{in}x_{in})$.

Instead of minimizing the sum objective function, OWA TSP optimizes the OWA objective function $\sum_{i\in[n]} w_i\theta_i(\boldsymbol{v})$ which can encode both the efficiency and fairness of $\boldsymbol{v}$. A MIP formulation for OWA TSP based on the subtour polytope [2] can be written as follows:

$$\begin{aligned} &\min \sum_{k\in[n]} w_k\theta_k(\boldsymbol{v}) && && \text{(3a)}\\ &\text{s.t } v_k=\sum_{i\in[n]} c_{ik}x_{ik} && \forall k\in[n] && \text{(3b)}\\ &\sum_{i\in[n]} x_{ik}=1 && \forall k\in[n] && \text{(3c)}\\ &\sum_{i\in[n]} x_{ki}=1 && \forall k\in[n] && \text{(3d)}\\ &\sum_{i,j\in Q,i\neq j} x_{ij}\le|Q|-1 && \forall Q\subset V && \text{(3e)}\\ &x_{ik}\in\{0,1\} && \forall i,k\in[n] && \text{(3f)}\end{aligned}$$

In (3), constraints (3b) correspond to constraint $\boldsymbol{v}=\boldsymbol{C}\boldsymbol{x}$ and constraints (3c), (3d) and (3e) represent explicitly constraint $\boldsymbol{A}\boldsymbol{x}\le\boldsymbol{b}$ in $(Fair-\mathcal{P})$. OWA TSP is NP-hard since it is reduced from the finding a Hamiltonian cycle problem.

Because of the objective function, OWA-combinatorial optimization is non-linear. More precisely, the ordering operator θ in the OWA function causes OWA-combinatorial optimization to become non-linear even if original constraints are linear. Thankfully, the OWA function can be linearized by two methods studied in [4] and [1], which are presented now.

B. Formulation O-MIP [4]

In [4], Ogryczak et al. proposed a method to linearize the OWA function using the cumulative vector of $\theta(\boldsymbol{v})$. Denote $\overline{\theta}(\boldsymbol{v})$ the cumulative ordered vector of $\boldsymbol{v}$, i.e., $\overline{\theta}_k(\boldsymbol{v})=$

$\sum_{i\in[k]}\theta_i(\boldsymbol{v})$. Obviously, $\overline{\theta}_k(\boldsymbol{v})$ is the sum of k largest components of $\boldsymbol{v}$. When fixing $\boldsymbol{v}$, $\overline{\theta}_k(\boldsymbol{v})$ can be computed as a solution of a knapsack problem:

$$\overline{\theta}_k(\boldsymbol{v}) = \max\left\{\sum_{i=1}^{k} v_i a_{ki} \,|\, a_k \in [0,1]^n, \sum_{i=1}^{k} a_{ki} = k\right\} \tag{4}$$

To integrate into a LP where $\boldsymbol{v}$ is also variable, we take a dual of (4), which is

$$\begin{aligned} \overline{\theta}_k(\boldsymbol{v}) = \min \quad & k r_k + \sum_{i=1}^{n} d_{ki} \\ \text{s.t} \quad & r_k + d_{ki} \geq v_i \quad && \forall i \in [n] \\ & d_{ki} \geq 0 && \forall i \in [n]. \end{aligned} \tag{5}$$

Moreover, by setting $w'_i = w_i - w_{i+1}$ for $i \in [n-1]$ and $w'_n = w_n$, the objective function of $(Fair-\mathcal{P})$ can be rewritten as $\min \sum_{k\in[n]} w'_k \overline{\theta}_k(\boldsymbol{v})$. Combining with (5), we get a MIP formulation for $(Fair-\mathcal{P})$, called O-MIP:

$$\text{(O-MIP)} \quad \begin{aligned} \min \; & \sum_{k\in[n]} w'_k (k r_k + \sum_{i\in[n]} d_{ki}) && && \text{(6a)} \\ \text{s.t } & r_k + d_{ki} \geq v_i && \forall k, i \in [n] && \text{(6b)} \\ & d_{ki} \geq 0 && \forall k, i \in [n] && \text{(6c)} \\ & \boldsymbol{v} = \boldsymbol{C}\boldsymbol{x} && && \text{(6d)} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} && && \text{(6e)} \\ & \boldsymbol{x} \in \{0,1\}^m && && \text{(6f)} \end{aligned}$$

Before presenting another formulation, we exploit O-MIP's structure to get useful information for later analysis. We consider a dual of O-MIP's continuous relaxation (denoted as $(\mathcal{D}_O)$) which has the following form:

$$(\mathcal{D}_O) \quad \begin{aligned} \max \; & -\boldsymbol{u}^T\boldsymbol{b} - \sum_{k\in[m]} t_k && && \text{(7a)} \\ \text{s.t. } & \sum_{i\in[n]} y_{ki} = k w'_k && \forall k \in [n] && \text{(7b)} \\ & y_{ki} \leq w'_k && \forall k, i \in [n] && \text{(7c)} \\ & \sum_{k\in[n]} y_{ki} + z_i = 0 && \forall k \in [n] && \text{(7d)} \\ & (\boldsymbol{u}\boldsymbol{A})_k - (\boldsymbol{z}\boldsymbol{C})_k + t_k \geq 0 && \forall k \in [m] && \text{(7e)} \\ & y_{ki} \geq 0 && \forall k, i \in [n] && \text{(7f)} \\ & t_k \geq 0 && \forall k \in [m] && \text{(7g)} \\ & u_k \geq 0 && \forall k \in [K]. && \text{(7h)} \end{aligned}$$

Interestingly, if we fix $\boldsymbol{y}$ and take a dual of $(\mathcal{D}_O)$, we obtain a continuous relaxation of $(Min-\mathcal{P})$ with modified costs, i.e.

$$(RP_{\boldsymbol{y}}) \quad \begin{aligned} \min \; & \sum_{i\in[n]} (\sum_{k\in[n]} y_{ik}) v_i && \text{(8a)} \\ \text{s.t. } & \boldsymbol{v} = \boldsymbol{C}\boldsymbol{x} && \text{(8b)} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} && \text{(8c)} \\ & \boldsymbol{x} \in [0,1]^m && \text{(8d)} \end{aligned}$$

As a consequence, a solution $(\boldsymbol{x}, \boldsymbol{v})$ for the formulation $(RP_{\boldsymbol{y}})$ with integrality conditions $x_i \in \{0,1\}$ is also feasible for O-MIP (in the sense that one can also find $(\boldsymbol{r}, \boldsymbol{d})$ such that $(\boldsymbol{r}, \boldsymbol{d}, \boldsymbol{x}, \boldsymbol{v})$ is feasible for O-MIP and its corresponding objective function is equal to $\text{OWA}_{\boldsymbol{w}}(\boldsymbol{v})$). We denote $(P_{\boldsymbol{y}})$ the discrete version of $(RP_{\boldsymbol{y}})$ where $\boldsymbol{x} \in \{0,1\}^m$.

C. Formulation C-MIP [1]

An alternative approach to linearize $\text{OWA}_{\boldsymbol{w}}(\boldsymbol{v})$, studied in [1], starts from the observation that if we permute $\boldsymbol{w}$ and take the inner product with $\boldsymbol{v}$, $\text{OWA}_{\boldsymbol{w}}(\boldsymbol{v})$ is the maximum value of this inner product considering all permutations of $\boldsymbol{w}$. Formally, $\text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}) = \max_{\boldsymbol{w}_\tau \in \Pi} \boldsymbol{w}_\tau^\top \boldsymbol{v}$ where Π is the set of all permutations (of the coefficients) of the vector $\boldsymbol{w}$. Thus, with $\boldsymbol{v}$ fixed, the OWA function can be computed by the following LP problem:

$$\begin{aligned} \max \; & \sum_{i\in[n]} \sum_{k\in[n]} p_{ik} w_i v_k && && \text{(9a)} \\ \text{s.t } & \sum_{i\in[n]} p_{ik} = 1 && \forall k \in [n] && \text{(9b)} \\ & \sum_{k\in[n]} p_{ik} = 1 && \forall i \in [n] && \text{(9c)} \\ & p_{ik} \geq 0 && \forall i, k \in [n]. && \text{(9d)} \end{aligned}$$

Take a dual of (9) and integrate it into $(Fair-\mathcal{P})$, we get the second MIP formulation for $(Fair-\mathcal{P})$, called C-MIP:

$$\text{(C-MIP)} \quad \begin{aligned} \min \; & \sum_{i\in[n]} (\alpha_i + \beta_i) && && \text{(10a)} \\ \text{s.t } & \alpha_i + \beta_k \geq w_i v_k && \forall i, k \in [n] && \text{(10b)} \\ & \boldsymbol{v} = \boldsymbol{C}\boldsymbol{x} && && \text{(10c)} \\ & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} && && \text{(10d)} \\ & \boldsymbol{x} \in \{0,1\}^m && && \text{(10e)} \end{aligned}$$

We also consider a dual of C-MIP's continuous relaxation (denoted as $(\mathcal{D}_C)$) as the previous section.

$$(\mathcal{D}_C) \quad \begin{aligned} \max \; & -\boldsymbol{u}^T\boldsymbol{b} - \sum_{k\in[m]} t_k && && \text{(11a)} \\ \text{s.t. } & \sum_{i\in[n]} p_{ik} w_i + z_k = 0 && \forall k \in [n] && \text{(11b)} \\ & (\boldsymbol{u}\boldsymbol{A})_k - (\boldsymbol{z}\boldsymbol{C})_k + t_k \geq 0 && \forall k \in [m] && \text{(11c)} \\ & \sum_{i\in[n]} p_{ik} = 1 && \forall k \in [n] && \text{(11d)} \\ & \sum_{k\in[n]} p_{ik} = 1 && \forall i \in [n] && \text{(11e)} \\ & p_{ik} \geq 0 && \forall i, k \in [n] && \text{(11f)} \\ & t_k \geq 0 && \forall k \in [m] && \text{(11g)} \\ & u_k \geq 0 && \forall k \in [K]. && \text{(11h)} \end{aligned}$$

Then a dual of the above formulation with $\boldsymbol{p}$ fixed is:

$$(RP_{\boldsymbol{p}}) \quad \min \sum_{i\in[n]} \left(\sum_{k\in[n]} w_k p_{ik} \right) v_i \tag{12a}$$
$$\text{s.t. } \boldsymbol{v} = \boldsymbol{C}\boldsymbol{x} \tag{12b}$$
$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{12c}$$
$$\boldsymbol{x} \in [0,1]^m \tag{12d}$$

We denote $(P_{\boldsymbol{p}})$ the integer version of $(RP_{\boldsymbol{p}})$ over $\boldsymbol{x}$ (i.e., the constraint (12d) is replaced by $\boldsymbol{x} \in \{0,1\}^m$). Similar to O-MIP, solving $(RP_{\boldsymbol{p}})$ with $\boldsymbol{x}$ discrete yields a feasible solution for C-MIP.

In comparison, C-MIP's size is smaller than O-MIP's because of the number of extra variables. More precisely, although both formulations use the same number of additional constraints (i.e., n^2), C-MIP utilizes only $2n$ new variables instead of n^2+n as in O-MIP. The formulations were compared experimentally for continuous optimization [1]. In this paper, we provide a theoretical analysis (Section III) in order to understand these formulations better, and we experimentally evaluate them for combinatorial optimization (Section V).

III. Theoretical Analysis

In this section, we show that the formulations are equivalent in terms of linear relaxations. Then, we give an estimation of the OWA value corresponding to optimal solutions of $(Min-\mathcal{P})$ based on that of $(Fair-\mathcal{P})$.

A. Relation Between The Formulations

We first introduce two notations. Let $\mathcal{Y}$ be the set of points that satisfy (7b) and (7c), and $\mathcal{P}$ be the set of points that satisfy (11d), (11e) and (11f).

The following theorem relates the problems $(P_{\boldsymbol{y}})$ and $(P_{\boldsymbol{p}})$.

Theorem 1: There exists a one-to-one correspondence $\phi : \mathcal{P} \to \mathcal{Y}$ such that $\forall \boldsymbol{p} \in \mathcal{P}$, Problems $(P_{\boldsymbol{p}})$ and $(P_{\phi(\boldsymbol{p})})$ have the same solutions.

This theorem directly follows from the following two lemmas, which show that the set of feasible solutions of $(P_{\boldsymbol{p}})$ for $\boldsymbol{p} \in \mathcal{P}$ and that for $(P_{\boldsymbol{y}})$ for $\boldsymbol{y} \in \mathcal{Y}$ coincide.

Lemma 1: $\forall \boldsymbol{p}^* \in \mathcal{P}$, $\exists \boldsymbol{y}^* \in \mathcal{Y}$ such that $\forall i \in [n], \sum_{k\in[n]} w_k p^*_{ik} = \sum_{k\in[n]} y^*_{ik}$.

Proof: It is sufficient to prove the result for the extreme points of polytope $\mathcal{P}$. As any other point $\boldsymbol{p'}$ is a convex combination of some extreme solutions $\boldsymbol{p^1}, \ldots, \boldsymbol{p^k}$, its counterpart $\boldsymbol{y'}$ can be obtained by the same convex combination of the counterparts $\boldsymbol{y^1}, \ldots, \boldsymbol{y^k}$ of $\boldsymbol{p^1}, \ldots, \boldsymbol{p^k}$.

Recall that the set of extreme points of $\mathcal{P}$ is exactly the set of permutations on set $[n]$. Let $\boldsymbol{p}^*$ be an extreme point of $\mathcal{P}$. Therefore, all the components of $\boldsymbol{p}^*$ are null except for some n components $p^*_{i_1 1} = p^*_{i_2 2} = \ldots = p^*_{i_n n} = 1$. The counterpart $\boldsymbol{y}^*$ of $\boldsymbol{p}^*$ can be built as follows:

$$\begin{pmatrix} y^*_{i_1 1} = w'_1 & y^*_{i_1 2} = w'_2 & \ldots & \ldots & \ldots & y^*_{i_1 n} = w'_n \\ y^*_{i_2 1} = 0 & y^*_{i_2 2} = w'_2 & \ldots & \ldots & \ldots & y^*_{i_2 n} = w'_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ y^*_{i_k 1} = 0 & y^*_{i_k 2} = 0 & \ldots & y^*_{i_k k} = w'_k & \ldots & y^*_{i_k n} = w'_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & y^*_{i_n n} = w'_n \end{pmatrix}$$

By considering any column $k \in [n]$, one can check: $\boldsymbol{y}^* \in \mathcal{Y}$, i.e., $\sum_{j\in[n]} y^*_{i_j k} = k w'_k$ and $y^*_{i_j k} \leq w'_k$. Moreover, by summing any row i_j for $j \in [n]$, one can check: $\sum_{k\in[n]} y^*_{i_j k} = w_j = \sum_{k\in[n]} w_k p^*_{i_j k}$. ∎

Lemma 2: $\forall \boldsymbol{y}^* \in \mathcal{Y}$, $\exists \boldsymbol{p}^* \in \mathcal{P}$ such that $\forall i \in [n], \sum_{k\in[n]} w_k p^*_{ik} = \sum_{k\in[n]} y^*_{ik}$.

Proof: Similarly to Lemma 1, it suffices to show the result for any extreme solution $\boldsymbol{y}$ in $\mathcal{Y}$. Let us consider the following transportation problem (TP) where the set of the supply nodes and the set of demand nodes are both $[n]$. For any $i, j \in [n]$, the supply at supply node i is equal to $\sum_{k\in[n]} y^*_{ik}$ and the demand of demand node j is equal to w_j. Note that the total supply is equal to the total demand, because $\sum_{i\in[n]} y^*_{ik} = k w'_k$, $\sum_{k\in[n]} \sum_{i\in[n]} y^*_{ik} = \sum_{k\in[n]} k w'_k = \sum_{i\in[n]} w_i$. For any feasible solution of TP, $\boldsymbol{\Phi} = (\Phi_{ik})_{i,k\in[n]}$ where Φ_{ik} is the value of the commodity transported from supply node i to demand node k, we can define a vector $\boldsymbol{p}$ as follows: $p_{ik} = \frac{\Phi_{ik}}{w_k}$ for all $i, k \in [n]$, which by definition satisfies:

$$\sum_{i\in[n]} p_{ik} = \sum_{i\in[n]} \frac{\Phi_{ik}}{w_k} = \frac{\sum_{i\in[n]} \Phi_{ik}}{w_k} = 1, \tag{13}$$

and

$$\sum_{k\in[n]} y^*_{ik} = \sum_{k\in[n]} \Phi_{ik} = \sum_{k\in[n]} w_k p_{ik}. \tag{14}$$

If $\boldsymbol{p}$ does not satisfy (*) $\sum_{k\in[n]} p_{ik} = 1$ for all $i \in [n]$, we will construct a sequence $\boldsymbol{\Phi^t}$ and its associated $\boldsymbol{p}^t$ such that $||\sum_{k\in[n]} \boldsymbol{p}^t_{\cdot k} - \boldsymbol{1}||_1 = \sum_{i\in[n]} |\sum_{k\in[n]} p^t_{ik} - 1|$ tends to zero, where $\boldsymbol{1}$ denotes the vector in $\mathbb{R}^n$ whose components are all equal to 1.

We first show that if $\boldsymbol{p}$ does not satisfy (*), then $\exists i, j, h, l \in [n]$ such that $h < l$ and $\Phi_{il} > 0$ and $\Phi_{jh} > 0$. Let $P_1 = \{i \in [n] \mid \sum_{k\in[n]} p^*_{ik} \geq 1\}$ and $P_2 = \{i \in [n] \mid \sum_{k\in[n]} p^*_{ik} < 1\}$ be a partition of $[n]$. By assumption, $P_2 \neq \emptyset$. Let $p_1 = |P_1|$, $p_2 = |P_2|$, $W_1 = [p_1]$, and $W_2 = \{p_1+1, p_1+2, \ldots, n\}$. Among the following four cases, only one is possible:

- There exist both strictly positive flows from supply nodes in P_1 to demand nodes in W_2 and strictly positive flows from supply nodes in P_2 to demand nodes in W_1. Thus, there are some $i \in P_1$, $j \in P_2$, $h \in W_1$ and $l \in W_2$ such that $\Phi^*_{il} > 0$ and $\Phi^*_{jh} > 0$.
- There exist strictly positive flows from supply nodes in P_1 to demand nodes in W_2 but no strictly positive flows from supply nodes in P_2 to demand nodes in W_1. Observe that the p_1 greatest values that $\sum_{k\in[n]} y^*_{ik}$ for $i \in [n]$ can take are $w'_1 + w'_2 + \ldots + w'_{n-1} + w'_n = w_1$, $w'_2 + w'_3 + \ldots + w'_n = w_2$, $\ldots$, $w'_{p_1} + w'_{p_1+1} + \ldots + w'_n = w_{p_1}$. Hence $\sum_{i\in P_1} \sum_{k\in[n]} y^*_{ik} \leq \sum_{i\in[p_1]} w_i$, i.e., the total supply in P_1 is less than or equal to the total demand in W_1. Thus, it is impossible that W_1 only receive flows from P_1 and P_1 can still send positive flows to W_2.
- There exist strictly positive flows from supply nodes in P_2 to demand nodes in W_1 but no strictly positive flows from supply nodes in P_1 to demand nodes in W_2. Thus, strictly

positive flows to W_2 come uniquely from P_2. Hence, we can observe that

$$\sum_{i\in P_2,k\in W_2} p^*_{ik} = \sum_{i\in[n],k\in W_2} \frac{\Phi^*_{ik}}{w_k} = \sum_{k\in W_2} \frac{w_k}{w_k} = |W_2| = p_2$$

But as $\sum_{k\in[n]} p^*_{ik} < 1$ for all $i \in P_2$, $\sum_{i\in P_2,k\in[n]} p^*_{ik} < |P_2| = p_2$, we have then

$$p_2 = \sum_{i\in P_2,k\in W_2} p^*_{ik} \le \sum_{i\in P_2,k\in[n]} p^*_{ik} < p_2,$$

which is contradictory.

- There is no strictly positive flows from supply nodes in P_2 to demand nodes in W_1 and no strictly positive flows from supply nodes in P_1 to demand nodes in W_2. This case is impossible due to the same reason as the previous one.

Let $\boldsymbol{\Phi}^0$ be a feasible solution of TP and $\boldsymbol{p}^0$ its associated vector. For any $t \in \mathbb{N}$, two cases can occur: (1) if $\boldsymbol{p}^t$ satisfies (*), $\boldsymbol{\Phi}^{t+1} = \boldsymbol{\Phi}^t$ and $\boldsymbol{p}^{t+1} = \boldsymbol{p}^t$; otherwise (2) $\boldsymbol{\Phi}^{t+1}$ and $\boldsymbol{p}^{t+1}$ can defined according to the following procedure. As (*) is not satisfied, there exist $i, j, h, l \in [n]$ such that $h < l$ and $\Phi^t_{il} > 0$ and $\Phi^t_{jh} > 0$. We then define $\Phi^{t+1} = \Phi^t$ except for the following terms: $\Phi^{t+1}_{ih} = \Phi^t_{ih} + \epsilon$, $\Phi^{t+1}_{il} = \Phi^t_{il} - \epsilon$, $\Phi^{t+1}_{jh} = \Phi^t_{jh} - \epsilon$ and $\Phi^{t+1}_{jl} = \Phi^t_{jl} + \epsilon$, where $\epsilon = \min(\Phi^t_{il}, \Phi^t_{jh}, \frac{(\sum_{k\in[n]} p^t_{ik}-1)w_l w_h}{w_h-w_l}, \frac{(1-\sum_{k\in[n]} p^t_{jk})w_l w_h}{w_h-w_l})$. Note that $\epsilon > 0$ since $w_h > w_l$.

Consequently, $\boldsymbol{p}^{t+1}$ satisfies: $p^{t+1}_{ih} = p^t_{ih} + \frac{\epsilon}{w_h}$, $p^{t+1}_{il} = p^t_{il} - \frac{\epsilon}{w_l}$, $p^{t+1}_{jh} = p^t_{jh} - \frac{\epsilon}{w_h}$ and $p^{t+1}_{jk} = p^t_{jl} + \frac{\epsilon}{w_l}$. By construction, $\boldsymbol{\Phi}^{t+1}$ and $\boldsymbol{p}^{t+1}$ verify (13) and (14). Moreover, $\sum_{k\in[n]} p^{t+1}_{ik}$ is decreased by $\frac{\epsilon(w_l-w_h)}{w_h w_l}$ and the sum $\sum_{k\in[n]} p^*_{jk}$ is increased by the same quantity. Hence, $||\sum_{k\in[n]} \boldsymbol{p}^t_{\cdot k} - \mathbf{1}||_1 > ||\sum_{k\in[n]} \boldsymbol{p}^{t+1}_{\cdot k} - \mathbf{1}||_1$. Clearly, sequence $(\boldsymbol{p}^t)$ converges to a vector in $\mathcal{P}$, which proves this lemma. ∎

B. Quality estimation for the optimal solution of $(Min - \mathcal{P})$

Due to the size of the formulations, MIP solvers can only solve small-sized instances within a reasonable amount of time. Thus, if the optimal solution of $(Min - \mathcal{P})$ is "good enough" for $(Fair - \mathcal{P})$, solving exactly large-sized instances is needless. The following theorem provides an estimation to evaluate the quality of $(Min - \mathcal{P})$'s optimal solution.

Theorem 2: Assume that there exists an approximation ratio r between $(Min - \mathcal{P})$ and its continuous relaxation $(Min - \mathcal{RP})$. Let $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{v}})$ be the optimal solution of $(Min - \mathcal{P})$ and $C = r \min\left(\frac{nw_1}{\sum_{i=1}^n w_i}, \frac{n\theta_1(\overline{\boldsymbol{v}})}{\sum_{i\in[n]} \overline{v}_i}\right)$, we have:

$$\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) \le C \times \text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}^*) \tag{15}$$

where $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is the optimal solution of $(Fair - \mathcal{P})$.

Proof: In this proof, we will use $OPT(\mathcal{A})$ to denote the optimal objective value of problem $\mathcal{A}$ (for example, $OPT(\mathcal{D}_O)$ is the optimal objective value of $(\mathcal{D}_O)$). From the assumption, we have:

$$OPT(Min - \mathcal{P}) \le rOPT(Min - \mathcal{RP}) \tag{16}$$

To establish the result (15), we only need to prove that:

$$\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) \le rC_i \times \text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}^*), \qquad \forall i = 1, 2$$

where $C_1 = \dfrac{nw_1}{\sum_{i=1}^n w_i}$, $C_2 = \dfrac{n\theta_1(\overline{\boldsymbol{v}})}{\sum_{i\in[n]} \overline{v}_i}$.

The bound with C_1 results from C-MIP while that of C_2 comes from O-MIP.

1) Proof of $\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) \le rC_1 \times \text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}^*)$: Let $\overline{\boldsymbol{p}} \in \mathbb{R}^{n\times n}$ subject to $\overline{p}_{ik} = 1/n$ for $i, k \in [n]$. Obviously, $\overline{\boldsymbol{p}}$ satisfies constraints (11d) - (11f) of $(\mathcal{D}_C)$. The objective value of $(RP_{\overline{\boldsymbol{p}}})$ is:

$$\sum_{i\in[n]} \left(\sum_{j\in[n]} w_j \overline{p}_{ij}\right) v_i = \frac{\sum_{j\in[n]} w_j}{n} \sum_{i\in[n]} v_i \tag{17}$$

Therefore, $OPT(P_{\overline{\boldsymbol{p}}}) = \frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{P})$ and $OPT(RP_{\overline{\boldsymbol{p}}}) = \frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{RP})$. This observation leads to:

$$\begin{aligned}
\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) &= \sum_{i\in[n]} w_i \theta_i(\overline{\boldsymbol{v}}) \\
&= \sum_{i\in[n]} \frac{nw_i}{\sum_{j\in[n]} w_j} \frac{\sum_{j\in[n]} w_j}{n} \theta_i(\overline{\boldsymbol{v}}) \\
&\le \frac{nw_1}{\sum_{j\in[n]} w_j} \left(\sum_{i\in[n]} \frac{\sum_{j\in[n]} w_j}{n} \theta_i(\overline{\boldsymbol{v}})\right) \\
&= C_1 \frac{\sum_{j\in[n]} w_j}{n} \sum_{i\in[n]} \overline{v}_i \\
&= C_1 \frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{P})
\end{aligned} \tag{18}$$

Using the relation between $OPT(Min - \mathcal{P})$ and $OPT(Min - \mathcal{RP})$, we get:

$$\begin{aligned}
&\frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{P}) \\
&\le r \frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{RP}) \\
&= rOPT(RP_{\overline{\boldsymbol{p}}}) \\
&\overset{(a)}{\le} rOPT(\mathcal{D}_C) \\
&\overset{(b)}{\le} rOPT(C - MIP) \\
&= r\text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}^*)
\end{aligned} \tag{19}$$

where (a) : when *fixing* $\boldsymbol{p} = \overline{\boldsymbol{p}}$, we have $OPT(\mathcal{D}_C) = OPT(RP_{\overline{\boldsymbol{p}}})$ (strong duality) and both are smaller than $OPT(\mathcal{D}_C)$ without fixing $\boldsymbol{p} = \overline{\boldsymbol{p}}$; (b): since $(\mathcal{D}_C)$ is a dual of C-MIP's continuous relaxation.

We obtain the proof by combining (18) and (19).

2) Proof of $\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) \leq rC_2 \times \text{OWA}_{\boldsymbol{w}}(\boldsymbol{v}^*)$: Recall that $w'_i = w_i - w_{i+1}$ for $i \in [n-1]$ and $w'_n = w_n$. From O-MIP, we have:

$$\begin{aligned}
\text{OWA}_{\boldsymbol{w}}(\overline{\boldsymbol{v}}) &= \sum_{k\in[n]} \left(\sum_{i\in[k]} \theta_i(\overline{\boldsymbol{v}}) \right) w'_k \\
&= \sum_{k\in[n]} \frac{n\sum_{i\in[k]} \theta_i(\overline{\boldsymbol{v}})}{k\sum_{i\in[n]} \overline{v}_i} \left(\frac{k}{n} \sum_{i\in[n]} \overline{v}_i \right) w'_k \\
&\leq \frac{n\theta_1(\overline{\boldsymbol{v}})}{\sum_{i\in[n]} \overline{v}_i} \sum_{k\in[n]} \left(\frac{k}{n} \sum_{i\in[n]} \overline{v}_i \right) w'_k \\
&= C_2 \left(\sum_{k\in[n]} \frac{k}{n} w'_k \right) \sum_{i\in[n]} \overline{v}_i \\
&= C_2 \frac{\sum_{j\in[n]} w_j}{n} \sum_{i\in[n]} \overline{v}_i \\
&= C_2 \frac{\sum_{j\in[n]} w_j}{n} OPT(Min - \mathcal{P})
\end{aligned} \tag{20}$$

The desired result follows by the application of (19) and (20).

■

Remark 1: Compared to the ratio established in [3], Theorem 2 generally has an additional factor r. This factor depends on the actual problem. For instance, if one considers the metric TSP where edge weights satisfy triangle inequalities, this factor will be 3/2 [6].

Theorem 2 provides a way to choose the weight $\boldsymbol{w}$ for $(Fair - \mathcal{P})$. Here are several typical choices of $\boldsymbol{w}$ and their corresponding C_1:

1) $w_i = (n-i+1)/n$: $C_1 = 2n/(n+1) < 1/2$.
2) $w_i = 1/i$: $C_1 = O(n/\log n)$ (since $\sum_{i\in[n]} 1/i \approx \log n$).
3) $w_i = 1/i^2$: $C_1 = O(n)$ (since $1 \leq \sum_{i\in[n]} 1/i^2 \leq \pi^2/6$)

In the last case where we choose $w_i = 1/i^2$ for $i \in [n]$, solving $(Fair - \mathcal{P})$ is necessary since the ratio C_1 is large.

IV. A Primal-Dual Heuristic

As presented in Section II, the MIP formulations are only efficient for small-sized instances. A primal-dual heuristic based on O-MIP is proposed in [3] to deal with larger-sized instances. In this section, we generalize this method for both formulations, sketched in Algorithm 1.

The algorithm starts from an initialization of $\boldsymbol{y}^{(0)}$ (resp. $\boldsymbol{p}^{(0)}$) satisfying conditions (7b), (7c), and (7f) (resp. (11d) - (11f)). For example, we can choose $\boldsymbol{y}^{(0)}$ (resp. $\boldsymbol{p}^{(0)}$) as proposed in Section III. Then $\boldsymbol{y}^{(t)}$ (resp. $\boldsymbol{p}^{(t)}$) is updated iteratively based on the improvement of lower bounds obtained from the Lagrangian relaxation corresponding to O-MIP (resp. C-MIP). For space reason, we focus on C-MIP. In detail, the

Algorithm 1

$t \leftarrow 0$
initialize $\boldsymbol{y}^{(0)}$ (resp. $\boldsymbol{p}^{(0)}$)
repeat
  $t \leftarrow t+1$
  solve $(P_{\boldsymbol{y}^{(t-1)}})$ (resp. $(P_{\boldsymbol{p}^{(t-1)}})$) to obtain a feasible solution $(\boldsymbol{v}^{(t)}, \boldsymbol{x}^{(t)})$
  update $\boldsymbol{y}^{(t)}$ (resp. $\boldsymbol{p}^{(t)}$) based on $\boldsymbol{y}^{(t-1)}$ (resp. $\boldsymbol{p}^{(t-1)}$) and $(\boldsymbol{v}^{(t)}, \boldsymbol{x}^{(t)})$
until max iteration has been reached or change on $\boldsymbol{y}^{(t)}$ (resp. $\boldsymbol{p}^{(t)}$) is small
return $(\boldsymbol{v}^{(t)}, \boldsymbol{x}^{(t)})$ with smallest OWA value

Lagrangian relaxation of C-MIP with respect to constraint (10b) can be defined as follows

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\lambda}) = \min & \sum_{i\in[n]} (1 - \sum_{j\in[n]} \lambda_{ij})\alpha_i + \sum_{j\in[n]} (1 - \sum_{i\in[n]} \lambda_{ij})\beta_i \\
& + \sum_{i\in[n]} \sum_{k\in[n]} \lambda_{ik} w_i v_k \\
\text{s.t. } & \boldsymbol{v} = \boldsymbol{C}\boldsymbol{x} && \text{(21a)} \\
& \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} && \text{(21b)} \\
& \boldsymbol{x} \in \{0,1\}^m && \text{(21c)}
\end{aligned}$$

where $\boldsymbol{\lambda} = (\lambda_{ik})_{i\in[n],k\in[n]}$ is a Lagrangian multiplier.

Solving (21) obtains a lower bound for C-MIP. The Lagrangian multiplier $\boldsymbol{\lambda}$ has to belong to set $\mathbb{L} = \{\boldsymbol{\lambda} \in \mathbb{R}_+^{n\times n} | \sum_{i\in[n]} \lambda_{ik} = 1 \ \forall k \in [n], \sum_{k\in[n]} \lambda_{ik} = 1 \ \forall i \in [n]\}$ to get a meaningful bound. Interestingly, if we decompose formulation (21) into two subproblems $(S1)$, $(S2)$ where $(S1)$ over $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $(S2)$ over $(\boldsymbol{v}, \boldsymbol{x})$, the subproblem (S_2) is exactly formulation $(P_{\boldsymbol{p}})$. Thus, updating $\boldsymbol{p}$ is equivalent to updating a projected sub-gradient step, i.e:

$$\begin{aligned}
\lambda'_{ij} &\leftarrow \lambda_{ij} - \gamma(\alpha_i + \beta_j - w_i v_j) \quad \forall i \in [n], k \in [n] && \text{(22a)} \\
\boldsymbol{\lambda} &\leftarrow \arg\min_{\boldsymbol{\lambda}\in\mathbb{L}} ||\boldsymbol{\lambda}' - \boldsymbol{\lambda}|| && \text{(22b)}
\end{aligned}$$

where γ is learning rate. Since the constraints of $\mathbb{L}$ involve with both rows and columns of $\boldsymbol{\lambda} \in \mathbb{R}^{n\times n}$, the projection on $\mathbb{L}$ cannot be solved by the capped simplex projection as in [3]. Hence, we perform the projection on $\mathbb{L}$ by minimizing a convex quadratic function over linear constraints.

V. Numerical results

In this section, we provide experimental results of two MIP formulations and Algorithm 1 for a specific combinatorial optimization problem, e.g., OWA TSP. We tested on several instances of TSPLIB [5] with the number of nodes in the range of 14 to 100. The weight $\boldsymbol{w}$ is defined as $w_k = 1/k^2$ for $k \in [n]$. We limited the solving time for each instance to three hours (10800s). All experiments are implemented in C++ programming language and conducted on a computer with 3GHz Intel Core i5 CPU and 16GB of RAM. We used

TABLE I
NUMERICAL RESULTS FOR OWA TSP.

Instance	O-MIP		C-MIP		$(\mathcal{H}_O)$		$(\mathcal{H}_C)$	
	CPU1(s)	**CPU2(s)**	**CPU1(s)**	**CPU2 (s)**	**CPU(s)**	**Gap(%)/Obj**	**CPU(s)**	**Gap(%)/Obj**
burma14	0.83	0.27	0.45	0.20	1.50	0.23%	1.62	0.06%
gr21	1.46	1.40	1.08	1.04	1.82	0.02%	2.25	0%
fri26	21.01	9.35	49.97	9.08	5.74	0%	4.12	0.02%
bays29	37.89	7.44	57.85	21.06	9.92	0.86%	11.10	1.19%
gr48	863.02	334.84	2419.39	354.91	57.17	2.86%	56.19	1.75%
hk48	1943.39	440.99	719.89	443.49	32.21	0.03%	31.14	0.48%
brazil58	TL	180.27	TL	664.29	21.85	(4770.24)	21.88	(4772.03)
st70	TL	TL	TL	TL	207.48	(35.89)	169.53	(35.97)
kroA100	TL	TL	TL	TL	1202.54	(811.42)	1657.44	(790.76)

ILOG CPLEX version 12.10.0 with default parameters and one thread to solve LP and MIP problems.

For simplicity, let $\mathcal{H}_O$, $\mathcal{H}_C$ be Algorithm 1 utilizing O-MIP and C-MIP, respectively. We initialized $y_{ik}^{(0)} = \frac{k}{n} w'_k$ and $p_{ik}^{(0)} = \frac{1}{n}$ for $i, k \in [n]$. At time step t, the learning rate $\gamma^{(t)}$ is computed as $\gamma^{(t)} = \frac{OPT(P) - \text{OWA}_{\boldsymbol{w}}(v')}{||\boldsymbol{\lambda}^{(t-1)}||_2}$ where $OPT(P)$ is the objective value of $(P_{\boldsymbol{y}^{(t-1)}})$ (resp. $(P_{\boldsymbol{p}^{(t-1)}})$), $\boldsymbol{v}'$ is the best feasible solution of O-MIP (resp. C-MIP) so far. Since all formulations for OWA TSP have an exponential number of subtour elimination constraints, we solve them by the branch-and-cut algorithm. In detail, when a feasible solution x^* is found, we construct a graph $G^* = (V, E^*)$ where $E^* = \{ij \in E^d | x^*_{ij} = 1\}$. Then a strongly connected component that does not contain all nodes of G^* corresponds to a subtour elimination constraint violated by x^*. Violated constraints (if any) are added to the formulations.

In Table I, the number in the instance's name is the number of nodes. Columns "O-MIP" and "C-MIP" regroup results of O-MIP and C-MIP, where subcolumn "CPU1" reports the time in seconds that CPLEX spent to obtain the optimal solution and subcolumn "CPU2" reports the time to obtain a feasible solution as small as the solution of $\mathcal{H}_O$ (resp. $\mathcal{H}_C$). Results of instances that can not be solved within the time limit (i.e., 10800s) are set to "TL" (stand for *Time Limit*). Columns "$\mathcal{H}_O$" and '$\mathcal{H}_C$" report respectively results of $\mathcal{H}_O$ and $\mathcal{H}_C$, where subcolumn "CPU" is the runtime and "Gap/Obj" provides the gap in percentage between the solutions of O-MIP (resp. C-MIP) and Algorithm 1. In the case that MIP formulations can not be solved within the time limit, we provide instead the objective value (numbers in parentheses, without the percent sign) corresponding to a solution found by $\mathcal{H}_O$ (resp. $\mathcal{H}_C$).

Numerical results confirm that MIP formulations can only handle small-sized instances. The running time spent to solve these formulations increases rapidly with instances' size and can reach up to around three hours. For example, instances with more than 50 vertices can not be solved within three hours. Interestingly, although C-MIP was shown to be better than O-MIP for OWA simplified portfolio optimization problem [1], its performance is worse than O-MIP's one when applying for OWA TSP. In contrast, $\mathcal{H}_O$ and $\mathcal{H}_C$ solve instance quickly, especially large-sized instances. The time spent by Algorithm 1 increases acceptably with instance size. Furthermore, their solutions are high-quality, namely that their optimality gap is at most 3%.

VI. CONCLUSION

In this paper, we study OWA-combinatorial optimization, a fair variant of combinatorial optimization which uses the OWA objective function. Our result extends the previous one to deal with OWA-combinatorial optimization. Our theoretical results show that two proposed schemes of OWA linearization in the literature are equivalent in terms of linear relaxations in the context of OWA-combinatorial optimization. These linearizations can also be exploited to compare the OWA values of optimal solutions of the original and fair version of a combinatorial optimization problem. Numerical results show that O-MIP can be solved faster, despite using more variables than C-MIP for OWA TSP. Our future work will focus on deriving more efficient heuristics and developing theoretical guarantees for those heuristics.

REFERENCES

[1] André Chassein and Marc Goerigk. Alternative formulations for the ordered weighted averaging objective. *Information Processing Letters*, 115(6-8):604–608, 2015.

[2] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

[3] Viet Hung Nguyen and Paul Weng. An efficient primal-dual algorithm for fair combinatorial optimization problems. In *International Conference on Combinatorial Optimization and Applications*, pages 324–339. Springer, 2017.

[4] Włodzimierz Ogryczak and Tomasz Śliwiński. On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research*, 148(1):80–91, 2003.

[5] Gerhard Reinelt. Tsplib—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[6] David B Shmoys and David P Williamson. Analyzing the held-karp tsp bound: A monotonicity property with application. *Information Processing Letters*, 35(6):281–285, 1990.