CNF Encodings for the Min-Max Multiple Traveling Salesmen Problem

Aolong Zha, Rongxuan Gao, Qiong Chang, Miyuki Koshimura and Itsuki Noda

November 11, 2020

CNF Encodings for the Min-Max Multiple Traveling Salesmen Problem

Aolong Zha*, Rongxuan Gao*, Qiong Chang*‡, Miyuki Koshimura† and Itsuki Noda*
*Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan
†Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan
*{aolong.zha, gao.ronshien, jou.kyuu, i.noda}@aist.go.jp, †koshi@inf.kyushu-u.ac.jp
‡Corresponding author

***Abstract*—In this study, we consider the multiple traveling salesmen problem (mTSP) with the min-max objective of minimizing the longest tour length. We begin by reviewing an existing integer programming (IP) formulation of this problem. Then, we present several novel conjunctive normal form (CNF) encodings and an approach based on modifying a maximum satisfiability (MaxSAT) algorithm for the min-max mTSP. The correctness and the space complexity of each encoding are analyzed. In our experiments, we compare the performance of solving the TSP benchmark instances using an existing encoding and our new encodings comparing the results achieved using an implemented group MaxSAT solver to those achieved using the IP method. The results show that for the same problem, the new encodings significantly reduce the number of generated clauses over the existing CNF encoding. Although the proposals are still not competitive compared to the IP method, one of them may be more effective on relatively large-scale problems, and it has an advantage over the IP method in solving an instance with a small ratio of the number of cities to the number of salesmen.**

***Index Terms*—Boolean satisfiability, min-max optimization, multiple traveling salesmen problem**

I. Introduction

As one of the classic problems in theoretical computer science, the traveling salesman problem (TSP) has also received much attention in operations research. Moreover, the TSP has been reformulated to address various practical application problems. The multiple TSP (mTSP) is a simple extension of TSP in which more than one salesman is deployed concurrently to visit a set of interconnected cities. All salesmen depart from and return to the same depot. Except for the depot, each of the cities can only be visited exactly once by a single salesman. A broader range of real-life problems can be modeled as the mTSP, including for example, mission planning [1], workload balancing [2], vehicle routing [3], etc. Regarding the objective to be optimized, there are two distinct directions, as follows: one that minimizes the total distance traveled by all the salesmen (*min-sum*), and another that minimizes the distance of the longest tour (*min-max*). Generally speaking, if we consider all tours as a vector, then the min-sum objective is to determine the minimal *Manhattan norm* of the vector, while the min-max objective is to determine its minimal *Chebyshev norm*. However, the min-sum objective conditionally results in highly imbalanced solutions in which one salesman visits all or most of the cities, if no restriction is imposed on the number of cities to be visited by each salesman. Furthermore, as an emphasis on practicality, multi agent cooperation does not aim to reduce costs, but rather to reduce the makespan to serve all the clients. For this reason, we focus in this paper on the min-max mTSP.

This paper is based on results obtained from a project commissioned by the New Energy and industrial technology Development Organization (NEDO). This work was also partially supported by JSPS KAKENHI Grant Number JP19H04175.

A. Related Work

From the standpoint of computational complexity theory, the mTSP is strongly $\mathcal{NP}$-hard as the TSP is a special case. In addition, for the min-max optimization problem, some theorems related to its computational complexity have been proved in the following literature. Yu [4] discussed the corresponding min-max version of several classical discrete optimization problems including the minimum spanning tree problem, the resource allocation problem, and the production control problem. The strong $\mathcal{NP}$-hardness of these problems is shown for an unbounded number of scenarios. Ko and Lin [5] presented a number of optimization problems, such as the min-max clique problem, the min-max three-dimensional matching problem, and the min-max circuit problem, and showed that they are complete for the class $\Pi_2^{\mathcal{P}}$, the second level of the polynomial-time hierarchy. Aissi et al. [6] proved that the min-max and min-max regret versions of the assignment problem are strongly $\mathcal{NP}$-hard when the number of scenarios is not bounded by a constant.

Because the min-max mTSP is much more difficult to solve than the min-sum version, only very small-scale instances of the min-max mTSP can be solved optimally within an appropriate time limit. One of the earliest exact algorithms for the min-max mTSP, discussed by França et al. [7], was based on the Tabu search heuristic with the dichotomous and downward search schemes. Despite the important academic and engineering value of the min-max mTSP, the research on it is relatively limited and different heuristic approaches have been developed in the literature. Frederickson et al. [8] proposed some approximation algorithms, which included k-near insert, k-near neighbor, and k-split tour for the min-max mTSP. Somhom et al. [9] and Modares et al. [10] developed a self-organizing neural network approach for the min-max mTSP, which introduced a competition method to decide whether a city should be included in a tour. Soylu [11] presented a general variable neighborhood search algorithm. Necula et al. [12] and Venkatesh and Singh [13] respectively proposed various swarm intelligence algorithms, such as the ant colony, the artificial bee colony, and the invasive weed optimizations for the min-max mTSP. More recently, Vandermeulen et al. [14] formulated combined task assignment and routing problems as the minimum Hamiltonian partition problem, which is equivalent to the min-max mTSP, and developed a heuristic algorithm for solving it.

B. Contributions

We propose three *conjunctive normal form* (CNF) encodings for the min-max mTSP in reference to the characteristics of the *integer programming* (IP) formulation. These three encodings are all to prevent subtours occurring in the solution of the problem; two are based on vertex potential constraints, and the third is based on reachability constraints. For each proposed encoding, we provide a complete proof of its correctness and space complexity. In addition, we propose a group *maximum satisfiability* (MaxSAT) algorithm to solve the encoded min-max mTSP. We also provide a comparison between an existing naive CNF encoding, our proposals and an IP approach on the instances of the TSP benchmark. Source code for our experiments is available at https://github.com/ReprodSuplem/MTSP/.

II. Preliminaries

The mTSP is defined on a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. The graph is associated with a distance matrix $D = (d_{ij})$ for each edge $(i, j) \in E$. The matrix D is said to be *symmetric* when $d_{ij} = d_{ji}, \forall (i, j) \in E$ and *asymmetric* otherwise.

A. IP Formulation for the Min-Max mTSP

Owing to the two-dimensional characteristics of edges, the min-sum mTSP is typically formulated using an assignment-based double-index IP formulation, while for the min-max mTSP, a general scheme is to add a third dimension in order to distinguish clearly among the edges assigned to each salesman. Therefore, we let x_{ijk} be a binary variable that is equal to 1 if edge (i, j) is selected in the k-th salesman's tour and 0 otherwise. We also define an integer variable u_i (the *potential* of vertex i) as the number of cities visited on a salesman's path from the depot to city i. Then, the min-max mTSP can be described as follows [12]:

$$\min \theta \tag{1}$$

$$\text{s.t.} \sum_{j=2}^{n} x_{1jk} = 1, \qquad 1 \le k \le m \tag{2}$$

$$\sum_{i=2}^{n} x_{i1k} = 1, \qquad 1 \le k \le m \tag{3}$$

$$\sum_{\substack{i=1\\ i\neq j}}^{n} \sum_{k=1}^{m} x_{ijk} = 1, \qquad 2 \le j \le n \tag{4}$$

$$\sum_{\substack{j=1\\ j\neq i}}^{n} \sum_{k=1}^{m} x_{ijk} = 1, \qquad 2 \le i \le n \tag{5}$$

$$\sum_{\substack{i=1\\ i\neq j}}^{n} x_{ijk} = \sum_{\substack{i=1\\ i\neq j}}^{n} x_{jik}, \qquad 2 \le j \le n,\ 1 \le k \le m \tag{6}$$

$$u_i - u_j + (n-m) \cdot \sum_{k=1}^{m} x_{ijk} \le n - m - 1, \qquad 2 \le i \ne j \le n \tag{7}$$

$$\sum_{(i,j)\in E} d_{ij} x_{ijk} \le \theta, \qquad 1 \le k \le m \tag{8}$$

$$x_{ijk} \in \{0,1\},\ \forall (i,j) \in E, \qquad 1 \le k \le m$$

where the number of cities (including the depot) is n, the number of salesmen is m with $m \le n$. Constraint (2) (resp., constraint (3)) guarantees that, for each salesman k, the depot (i.e., vertex 1) is to be departed from (resp., returned to) exactly once. Constraint (4) (resp., constraint (5)) ensures that each non-depot city is to be visited (resp., departed from) exactly once. Constraint (6) enforces that for each non-depot city, the salesman who enters and exits the same city must be consistent. Constraint (7) [15], [16] is based on the *subtour elimination constraint* (SEC) proposed by Miller et al. [17], referred to here as the MTZ-based SEC, where the generated formulae and the required vertex potentials are $O(n^2)$. This constraint is used to prevent *subtours*, which are degenerate tours that are formed between non-depot cities and not connected to the depot. In addition, this constraint ensures that each salesman is to visit at least one non-depot city. Here $n-m$ is the maximum number of vertices that can be visited by any salesman. The potential of each vertex indicates the order of the corresponding vertex in the tour. The objective function (1) is to minimize the auxiliary variable θ ($\theta \in \mathbb{R}$) indicating the upper bound of each salesman's tour length, as shown in inequality (8).

Definition 1. The min-max optimization problem (MMOP) is defined generally as follows:

$$\min_{k,\epsilon} \theta \quad \text{s.t. } \mathcal{C} \wedge \bigwedge_{k=1}^{m} \Big(f(k,\epsilon) \le \theta\Big), \tag{9}$$

where $f(k, \epsilon)$ is the individual cost function for k ($1 \le k \le m$), ϵ is a set of other related variables and $\mathcal{C}$ is the set of remaining constraints. Specially, for the min-max mTSP, $f(k, \epsilon) = \sum_{\epsilon \in E} d_\epsilon x_{\epsilon k}$, and $\mathcal{C}$ consists of constraints (2)–(7).

B. Maximum Satisfiability

A well-known *Boolean satisfiability problem* (SAT) was the first problem shown to be $\mathcal{NP}$-complete [18]; this problem requires determining whether there exists a truth assignment that satisfies a given Boolean formula.[1] Typically, a Boolean formula is expressed in CNF, consisting of a conjunction (using the symbol $\wedge$) of one or more clauses. A *clause* is a disjunction (using the symbol $\vee$) of one or more literals, and a *literal* is

[1] A truth assignment is a function $\mathcal{A} : X \to \{0, 1\}$, where X is a set of Boolean variables and $\mathcal{A}$ is regarded as a conjunction of all elements in X.

an occurrence of a Boolean variable or its negation (using the symbol $\neg$).

MaxSAT is an optimal version of SAT [19]. In the *weighted partial* MaxSAT, the problem instance is typically expressed as a set of hard and soft clauses, where each soft clause has a bounded positive numerical *weight* that indicates the cost of falsifying the soft clause. The problem is to find a model that satisfies all the hard clauses and minimizes the total cost (i.e., maximizing the sum of the weights of the satisfied soft clauses). Formally, we denote a MaxSAT formula as $\mathcal{F} = \mathcal{H} \wedge (C_{n+1}, w_1) \wedge \ldots \wedge (C_{n+m}, w_m)$, where $\mathcal{H}$ is the set of hard clauses consisting of $\bigwedge_{i=1}^{n} C_i$ and the remaining clauses (i.e., $\bigwedge_{i=1}^{m} C_{n+i}$) are soft. Solving a MaxSAT instance $\mathcal{F}$ amounts to finding an assignment that satisfies $\mathcal{H}$ and minimizes $\sum_{i=1}^{m}(w_i \neg C_{n+i})$. Technically, we introduce the auxiliary Boolean variable b_i for each soft clause C_{n+i} with the implication of $\neg b_i \rightarrow C_{n+i}$ where $1 \leq i \leq m$, and such a b_i is called the *blocking variable* [20], to ensure that $\mathcal{F}$ can be solved through the resolution of a sequence of SAT instances associated with the *pseudo-Boolean* (PB) constraints encoding [21] as follows:

$$\mathcal{F}_t = \mathcal{H} \wedge \Big(\bigwedge_{i=1}^{m} (C_{n+i} \vee b_i) \Big) \wedge \underset{\text{PB}}{\mathsf{CNF}} \Big(\sum_{i=1}^{m} w_i b_i < t \Big). \quad (10)$$

In (10), $\mathcal{F}_t$ is a CNF that is satisfiable if and only if $\mathcal{F}$ has an assignment $\mathcal{A}$ whose cost (i.e., $\sum_{i=1}^{m}(w_i \neg C_{n+i})$) is less than t. If the optimal assignment of $\mathcal{F}$ is $\mathcal{A}^*$ and its minimal cost is t^*, then the SAT problem $\mathcal{F}_t$ for $t \geq t^*$ is satisfiable, while the problem for $t < t^*$ is unsatisfiable. For SAT testing a sequence of $\mathcal{F}_t$, t is initialized to $\sum_{i=1}^{m} w_i + 1$, with the next t depending on the current assignment $\mathcal{A}$, obtained from the previous testing. Whenever $\mathcal{F}_t$ is unsatisfiable, t^* is the last tested and satisfiable t. Therefore, to search for the minimal cost for $\mathcal{F}$ is to find the precise location of this transition from satisfiable to unsatisfiable CNF formulae. This approach is typically called *SAT-based* MaxSAT algorithm [22].

C. Group MaxSAT

The MMOP shown in (9) can be transformed into general MaxSAT. However, such a reduction method might suffer from execution slow down or even memory-out caused by the huge size of the encoded formula. Owing to that, for each k, the constraint of $f(k, \epsilon) \leq \theta$ within the MMOP can be modeled as a group of soft clauses, the resulting formalism is referred to as *group MaxSAT* [23]. In group MaxSAT, each grouped soft clause can be regarded as a triple (C_{n+i}, w_i, k) indicating that the soft clause C_{n+i} corresponds to the weight w_i and is labeled as the k-th group of soft clauses. This modification allows us to distinguish between soft clauses in accordance with their respective groups.

Definition 2. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n Boolean variables. The following naive CNF encodings correspond to three special cases of cardinality constraints for X.

- At most one constraint: $AMO(X) = \bigwedge_{i=1}^{n} \bigwedge_{j=i+1}^{n} (\neg x_i \vee \neg x_j)$.
- At least one constraint: $ALO(X) = \bigvee_{i=1}^{n} x_i$.
- Exactly-one constraint: $EO(X) = AMO(X) \wedge ALO(X)$.

III. CNF Encodings for the MTZ-Based SEC

With the min-max mTSP in IP formulation, constraints (2)–(5) can be encoded directly into CNF by using the exactly-one constraint. In constraint (6), according to the previous constraints, we know that both $\sum_{i=1}^{n} x_{ijk}$ and $\sum_{i=1}^{n} x_{jik}$ are less than or equal to one. Therefore, the equation $\sum_{i=1}^{n} x_{ijk} = \sum_{i=1}^{n} x_{jik}$ is equivalent to its logical form $\bigvee_{i=1}^{n} x_{ijk} \leftrightarrow \bigvee_{i=1}^{n} x_{jik}$, which can also be represented simply as CNF. For encoding $\mathcal{C}$ in (9), the remaining work is to convert the MTZ-based SEC, viz., constraint (7), into a CNF formula.

A. Arithmetic Encoding

The most explicit method is to encode the arithmetic formula expressed as constraint (7) directly into CNF. For each vertex potential u_i, we have $0 \leq u_i \leq n-2$, where $2 \leq i \leq n$ and n is the number of cities. Therefore, according to the scheme of the constraint satisfaction problem (CSP) based on direct encoding [24], in total $(n-1)(n-2)$ Boolean variables are required corresponding to every possible value of all vertex potentials. We denote these Boolean variables as μ_{it}, where $2 \leq i \leq n$ and $1 \leq t \leq n-2$. Then constraint (7) can be rewritten in the following form:

$$\begin{aligned} \forall i, j,\ & \sum_{t=1}^{n-2} \Big(t(\mu_{it} - \mu_{jt}) \Big) + \sum_{k=1}^{m} \Big((n-m) x_{ijk} \Big) \\ & \leq n - m - 1 \\ \Leftrightarrow\ & \sum_{t=1}^{n-2} (t\mu_{it}) + \sum_{t=1}^{n-2} (t \neg \mu_{jt}) + \sum_{k=1}^{m} \Big((n-m) x_{ijk} \Big) \\ & \leq \frac{n^2 - n}{2} - m. \end{aligned} \quad (11)$$

Equation (11) is a canonical PB constraint that can be encoded into a CNF formula. The complexity of a state-of-the-art PB constraint encoding is $O(N^3 \log A)$, where N is the number of terms and A is the maximum of coefficients [25]. In the mTSP, we have $\binom{n-1}{2}$ such PB constraints and $\forall i$, $EO(\bigcup_{t=0}^{n-2} \mu_{it})$, where $N = 2(n-2)+m$, $A = n-2$ and $2 \leq i \leq n$. Therefore, the arithmetic encoding results in $O(n^5 + m^3 n^2)$ generated clauses.

B. Potential Encodings

In addition to encoding the arithmetic expression of constraint (7) directly, we can also achieve the logical conversion according to the specific meaning of the MTZ-based SEC. For any two distinct vertex potentials, the difference between their values must be less than or equal to $n-m-1$. Moreover, $\sum_{k=1}^{m} x_{ijk} = 1$ if and only if vertex j is adjacent to vertex i in the graph. Thus, in this case, we have a pair of vertex potentials for which $u_j - u_i \geq 1$, indicating that vertex j appears after vertex i in the permutation. In brief, the MTZ-based SEC acts to restrict the order of each pair of vertices in its tour if one such pair is the successor of the other one. Obviously, a solution will contradict the exhaustive MTZ-based SECs if it includes any subtour.

1) Guiding potential: We use the properties of vertex potentials described above to introduce a new type of Boolean variables ν_{ijt} to avoid the occurrence of subtours, with $2 \leq i \neq j \leq n$, $(i,j) \in E$, and t $(1 \leq t \leq n-2)$ indicating the potential of the successor vertex j. For example, $\nu_{ijt} = 1$ indicates that a salesman will go directly to city j when departing from city i which is the salesman's t-th arrival non-depot city. Therefore, almost $(n-2)(n-1)^2$ additional Boolean variables are required for the following encoding.

$$\bigwedge_{k=1}^{m} \Big(x_{ijk} \rightarrow \bigvee_{t=1}^{n-2} \nu_{ijt} \Big), \qquad 2 \leq i \neq j \leq n \quad (12)$$

$$\bigwedge_{t=1}^{n-2} \Big(\nu_{ijt} \rightarrow \bigvee_{k=1}^{m} x_{ijk} \Big), \qquad 2 \leq i \neq j \leq n \quad (13)$$

$$\bigwedge_{j=2}^{n} \Big(x_{1jk} \rightarrow x_{j1k} \vee \bigvee_{\substack{l=2 \\ l \neq j}}^{n} \nu_{jl1} \Big), \qquad 1 \leq k \leq m \quad (14)$$

$$\bigwedge_{t=1}^{n-2} \Bigg(\nu_{ijt} \rightarrow \bigvee_{k=1}^{m} x_{j1k} \vee \begin{cases} \bigvee_{l=2,\, l \neq j}^{n} \nu_{jl(t+1)}, & \text{if } t \neq n-2, \\ 0, & \text{otherwise.} \end{cases} \Bigg), \qquad 2 \leq i \neq j \leq n \quad (15)$$

$$\bigwedge_{j=2}^{n} AMO\Big(\bigcup_{t=1}^{n-2} \bigcup_{i=2}^{n} \nu_{ijt} \Big). \quad (16)$$

Equation (12) (resp., (13)) represents the implication from x_{ijk} to $\bigvee_{t=1}^{n-2} \nu_{ijt}$ (resp., from ν_{ijt} to $\bigvee_{k=1}^{m} x_{ijk}$). Equation (14) and (15) both guide x_{j1k} or ν_{jlt} in the forward direction of each tour (i.e., the next vertex potential), with the former corresponding to the potential value of 1 and the latter corresponding to the remaining cases. Equation (16) imposes the restriction that each successor vertex j can correspond only to at most one specific potential.

Theorem 1. The simultaneous (12)–(16) ensure that there are no subtours in the solution of the min-max mTSP.

Proof. Assume that an assignment $\mathcal{A}$ includes at least one subtour and that the set of edge connections in such subtour is $\{x_{ab\kappa}, \ldots, x_{ca\kappa}\}$ (i.e., $\mathcal{A}(x_{ab\kappa} \wedge \ldots \wedge x_{ca\kappa}) = 1$), where $2 \leq a \neq b \leq n$ and $2 \leq c \neq a \leq n$. This subtour indicates that the salesman κ travels from non-depot city a to b and finally returns to a from c. In accordance with (12), the projection relation from x_{ijk} to $\bigvee_{t=1}^{n-2} \nu_{ijt}$ is constructed, to ensure that $\bigvee_{t=1}^{n-2} \nu_{ijt}$ can reflect the edge connections of x_{ijk}. Since each route can be regarded as a series of head-to-tail edges with an increasing potential to each tail vertex, we begin by constraining the connection between the first arriving non-depot city and successor city through (14). The subsequent edge connections are guided according to (15). An inevitable contradiction will occur on the constraint of the last edge of the subtour. We have the following cases when $\mathcal{A}(\nu_{cat}) = 1$:

- If $t \neq n-2$, then $\nu_{cat} \rightarrow \bigvee_{k=1}^{m} x_{a1k} \bigvee_{l=2}^{n} \nu_{al(t+1)}$.
  - If $\mathcal{A}(\bigvee_{k=1}^{m} x_{a1k}) = 1$, then the definition of subtour is violated if $\mathcal{A}(x_{a1\kappa}) = 1$; inconsistencies with constraint (6) arise otherwise.
  - If $\mathcal{A}(\bigvee_{l=2}^{n} \nu_{al(t+1)}) = 1$, then a contradiction with (16) arises if $\mathcal{A}(\nu_{ab(t+1)}) = 1$; otherwise, according to (13), this conflicts with $\mathcal{A}(\bigvee_{k=1}^{m} x_{alk}) = 1$ $(l \neq b)$ for the same reason as the previous subitem whether $k = \kappa$ or not.
- Otherwise, we have $\nu_{ca(n-2)} \rightarrow \bigvee_{k=1}^{m} x_{a1k}$. This might contradict the definition of subtour or another constraint for the same reason as that mentioned in the first subitem of the previous item.

Consequently, solutions of min-max mTSP without any subtour are guaranteed by simultaneous (12)–(16). □

Theorem 2. Equations (12)–(16) always produce a polynomial-sized CNF that includes the number of generated clauses of complexity $O(n^5 + mn^2)$, where m is the number of salesmen and n is the number of cities.

Proof. The number of clauses involved in (12) is bounded above by $m(n-1)^2$, while that involved in (13) is bounded by $(n-2)(n-1)^2$. In (14) and (15), the number of clauses is bounded, respectively, by $m(n-1)$ and $(n-2)(n-1)^2$. Last, bounded by $(n-1)\binom{(n-2)(n-1)}{2}$ binary clauses are generated in (16). Therefore, (12)–(16) always produce a polynomial-sized CNF that includes the number of generated clauses of complexity $O(n^5 + mn^2)$, in which the number of binary clauses is $O(n^5)$. □

2) Blocking cycle: An alternative interpretation of MTZ-based SEC is that constraints prevent cycles with respect to the set of vertices $\{2, \ldots, n\}$. We still use the aforementioned new type of Boolean variables ν_{ijt}, but we replace the vertex potential guidance and some related constraints including (13)–(16) with the following constraints to block cycles.

$$\bigwedge_{i=2}^{n} \Big(\bigvee_{\substack{j=2 \\ j \neq i}}^{n} \nu_{ijt} \rightarrow \bigwedge_{\tau=t}^{n-2} \bigwedge_{\substack{l=2 \\ l \neq i}}^{n} \neg \nu_{li\tau} \Big), \qquad 1 \leq t \leq n-2 \quad (17)$$

Equation (17) indicates that, for potentials τ greater than or equal to the current potential t, the head of any edge connection (i.e., departure vertex) i can not be the tail of any edge connection (i.e., successor vertex).

Theorem 3. The simultaneous (12) and (17) guarantee that there are no subtours in the solution of the min-max mTSP.

Proof. Assume that an assignment $\mathcal{A}$ includes at least one subtour and consider that the set of edge connections in such a subtour is $S = \{x_{ab\kappa}, \ldots, x_{ca\kappa}\}$ (i.e., $\mathcal{A}(x_{ab\kappa} \wedge \ldots \wedge x_{ca\kappa}) = 1$), where $2 \leq a \neq b \leq n$ and $2 \leq c \neq a \leq n$. This subtour indicates that the salesman κ starts from non-depot city a to b and finally returns to a from c. According to (12), $\mathcal{A}(\bigvee_{t=1}^{n-2} \nu_{abt} \wedge \ldots \wedge \bigvee_{t=1}^{n-2} \nu_{cat}) = 1$. Now focus on $\mathcal{A}(\nu_{abt_1} \wedge \ldots \wedge \nu_{cat_{|S|}}) = 1$, where t_1 (resp., $t_{|S|}$) is the minimal t with respect to $\mathcal{A}(\bigvee_{t=1}^{n-2} \nu_{abt}) = 1$ (resp., $\mathcal{A}(\bigvee_{t=1}^{n-2} \nu_{cat}) = 1$) and $1 \leq t_1 < \ldots < t_{|S|} \leq n-2$ corresponding to the order of the assumed subtour's edge connections. Further according to (17), $\mathcal{A}(\bigwedge_{\tau=t_1}^{n-2} \bigwedge_{l=2}^{n} \neg \nu_{la\tau}) = 1$. From $\nu_{cat_{|S|}} \in \bigcup_{\tau=t_1}^{n-2} \bigcup_{l=2}^{n} \nu_{la\tau}$, it follows that $\mathcal{A}(\neg \nu_{cat_{|S|}}) = 1$, which

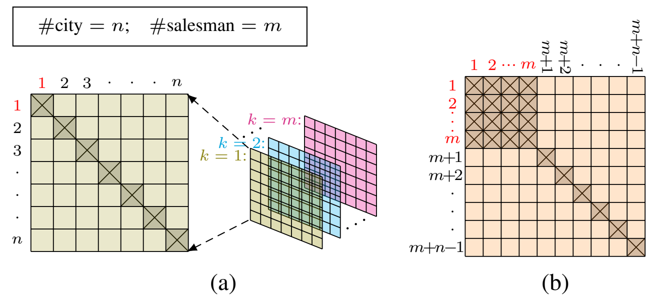

Fig. 1. Boolean variables of edge connections: (a) Each cell of a matrix indicates the variable x_{ijk} with the specific i (index of rows), j (index of columns), and k (index of salesmen), where for both rows and columns, index 1 corresponds to the depot city. (b) Each cell of the matrix indicates the variable l_{ij} with the specific i (index of rows) and j (index of columns), where for both rows and columns, indices from 1 to m, respectively, correspond to the depot city for each salesman k. The cells marked by ⊠ indicate that the corresponding variables can be omitted.

conflicts with the previous derivation. Therefore, solutions of min-max mTSP without any subtour can be ensured by (12) and (17). □

Theorem 4. Equations (12) and (17) always produce a polynomial-sized CNF that includes the number of generated clauses of complexity $O(n^5 + mn^2)$, where m is the number of salesmen and n is the number of cities.

Proof. The number of clauses required in (12) is bounded above by $m(n-1)^2$. In (17), the number of clauses is bounded by $\frac{(n-2)(n-1)^4}{2}$ and all these clauses are binary clauses. Therefore, (12) and (17) also produce a polynomial-sized CNF that includes the number of generated clauses of complexity $O(n^5 + mn^2)$, in which the number of binary clauses is of complexity $O(n^5)$. □

C. Relative Encoding

To prevent cycles other than the main tour for each salesman, we also propose an encoding method based on reachability constraints. In this encoding, we can compact the variables of edge connections further from triple-index to double-index. In contrast to the previous x_{ijk} shown in Fig. 1 (a), we define a new type of Boolean variables l_{ij} depicted in Fig. 1 (b). For the axes of i and j, instead of using the first index (highlighted red in Fig. 1 (a)) to represent the depot city in x_{ijk}, we use the first m indices (highlighted red in Fig. 1 (b)) to represent m duplications of the depot city sequentially for every salesmen in l_{ij}.

In addition, we also introduce another new type of Boolean variable r_{ij} which indicates whether vertex j can be reached via vertex i, where $1 \le i \le n$, $m+1 \le j \le m+n-1$, and $i \ne j$. In other words, $l_{ij} = 1$ if and only if vertex j appears immediately after vertex i in any salesman's tour; while $r_{ij} = 1$ if and only if vertex i appears before vertex j in any salesman's tour. This idea, based on the relative positions of vertices in the permutation, was first proposed in Prestwich [26] and was devoted to a CNF encoding of the *Hamiltonian path problem*. Therefore, $O(m^2 + n^2 + mn)$ Boolean variables are required for encoding $\mathcal{C}$ in (9), as follows.

$$\bigwedge_{i=1}^{m} EO\Big(\bigcup_{j=m+1}^{m+n-1} l_{ij}\Big), \tag{18}$$

$$\bigwedge_{j=1}^{m} EO\Big(\bigcup_{i=m+1}^{m+n-1} l_{ij}\Big), \tag{19}$$

$$\bigwedge_{i=m+1}^{m+n-1} EO\Big(\bigcup_{\substack{j=1\\ j\ne i}}^{m+n-1} l_{ij}\Big), \tag{20}$$

$$\bigwedge_{j=m+1}^{m+n-1} EO\Big(\bigcup_{\substack{i=1\\ i\ne j}}^{m+n-1} l_{ij}\Big), \tag{21}$$

$$\bigwedge_{i=1}^{m+n-1} \bigwedge_{\substack{j=m+1\\ j\ne i}}^{m+n-1} (l_{ij} \to r_{ij}), \tag{22}$$

$$\bigwedge_{i=m+1}^{m+n-1} \bigwedge_{j=i+1}^{m+n-1} (\neg r_{ij} \vee \neg r_{ji}), \tag{23}$$

$$\bigwedge_{i=1}^{m+n-1} \bigwedge_{j=m+1}^{m+n-1} \bigwedge_{k=m+1}^{m+n-1} (r_{ij} \wedge r_{jk} \to r_{ik}), \text{ condition set} \tag{24}$$

Equation (18) (resp., (19)) indicates that for each salesman, there exists exactly one departure from (resp., return to) the depot city to (resp., from) another city. Equation (20) (resp., (21)) specifies that each non-depot city can be departed from (resp., visited) exactly once. Equation (22) shows the implication from l_{ij} to r_{ij}. The acyclic constraints are given by (23). Last but not least, (24) indicates the transitive law of reachability variables, where the condition set includes that $k \ne j \ne i$ and among the three indices i, j, and k, at most one of them is less than or equal to m (i.e., indicating the depot city).

Because this encoding does not include an equation that corresponds to constraint (6), the consistency of a salesman who enters and exits the same city can be restricted only by reachability constraints. However, the Boolean variables r_{ij} range over $m+1 \le j \le m+n-1$. Consequently, the last edge connection of each salesman's tour that returns to the depot might not be able to maintain this consistency. For example, we might obtain a solution in which $\mathcal{A}(l_{\kappa a} \wedge l_{ab} \wedge \ldots \wedge l_{c\kappa'}) = 1$, indicating that the tour originates from salesman κ's depot to city a, then goes to the next cities $b, \ldots, c$, and finally returns to the depot of salesman κ'. Here $l_{c\kappa'}$ is the last edge connection in this tour. Such inconsistency does not affect the optimization results, since they have an identical distance cost (i.e., $d_{c\kappa} = d_{c\kappa'}$).

Theorem 5. The simultaneous (18)–(24) guarantee that except for the edge connection that returns the depot for each salesman, the edge connections are a partial solution of the min-max mTSP without any subtour.

Proof. Assume that an assignment $\mathcal{A}$ includes at least one subtour, consider that the set of edge connections in such subtour is

$\{l_{ab}, \dots, l_{ca}\}$, and suppose that they are both dominated by the salesman κ (i.e., $\mathcal{A}(l_{ab} \wedge \dots \wedge l_{ca} \wedge r_{\kappa a} \wedge \dots \wedge r_{\kappa c}) = 1$), where $m+1 \leq a \neq b \leq m+n-1$ and $m+1 \leq c \neq a \leq m+n-1$. According to (22), we have $\mathcal{A}(r_{ab} \wedge \dots \wedge r_{ca}) = 1$. Then according to (24), we can obtain $\mathcal{A}(r_{ac}) = 1$, and this contradicts (23). $\square$

Lemma 1. Equations (22)–(24) ensure that all reachability variables r_{ij} can be excluded from decision variables in the entire inference process of solving the min-max mTSP.

Proof. In Theorem 5, every reachability literal r_{ij} or $\neg r_{ij}$ involved in its proof are evaluated to 1 by unit propagations according to (22)–(24). Therefore, we can consider all r_{ij} as the support variables used to restrict the solution without any subtour and declare them to be non-decision variables. $\square$

Theorem 6. Equations (18)–(24) always produce a polynomial-sized CNF that includes a number of generated clauses of complexity $O(n^3 + mn^2)$, where m is the number of salesmen and n is the number of cities.

Proof. The number of clauses involved in (18) and (19) is bounded above by $2m\left(\binom{n-1}{2}+1\right)$, while that involved in (20) and (21) is bounded by $2(n-1)\left(\binom{m+n-1}{2}+1\right)$. In both (22) and (23) the number of clauses is bounded by $(n-1)(m+n-1)$, and in (24), it is bounded by $(n-1)^2(m+n-1)$. Therefore, (18)–(24) always produce a polynomial-sized CNF that includes a number of generated clauses of complexity $O(n^3+mn^2)$. $\square$

IV. Group MaxSAT Solving for the Min-Max mTSP

This section proposes an approach based on group MaxSAT for solving the min-max mTSP. We analyze the grouped soft clauses corresponding to our proposed encodings, and extend the weighted partial MaxSAT algorithm to handle such the group MaxSAT problems.

A. Grouped Soft Clauses

In the min-max mTSP, for the potential encodings, $f(k,\epsilon)$ $(1 \leq k \leq m)$ can be expressed as a linear adder, where $f(k,\epsilon) = \sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij} \cdot x_{ijk})$. While for the relative encoding, we must discuss the following two situations: (1) for the last edge connection (i.e., return to depot), the linear adder counts d_{ij} if $r_{ki} \wedge l_{ij} = 1$. Here, k is the index corresponding to the departure depot and j is the index corresponding to the back depot, where $1 \leq i \leq m+n-1$ and $1 \leq j, k \leq m$; (2) for the other edge connections, the linear adder counts d_{ij} if $r_{kj} \wedge l_{ij} = 1$, where $1 \leq i \leq m+n-1$, $m+1 \leq j \leq m+n-1$ and $1 \leq k \leq m$. Therefore, we have $f(k,\epsilon) = \sum_{i=1}^{m+n-1}\sum_{j=1}^{m}(d_{ij} \cdot r_{ki} \cdot l_{ij}) + \sum_{i=1}^{m+n-1}\sum_{j=m+1}^{m+n-1}(d_{ij} \cdot r_{kj} \cdot l_{ij})$ for the relative encoding. Note that, due to the difference between the directed graphs corresponding to the potential encodings and the relative encoding (see Fig. 1), the distance matrix $D = (d_{ij})$ for the relative encoding is an extension of that for the potential encodings. We translate the min-max mTSP into the group MaxSAT formulae respectively for the potential encodings and for the relative encoding as follows:

$$
\begin{aligned}
\underset{\text{(potential)}}{\mathcal{F}'} &= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}(\neg x_{ijk}, d_{ij}, k),\\
\underset{\text{(relative)}}{\mathcal{F}'} &= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\bigwedge_{i=1}^{m+n-1}\Big(\bigwedge_{j=1}^{m}(\neg r_{ki} \vee \neg l_{ij}, d_{ij}, k)\\
&\quad \bigwedge_{j=m+1}^{m+n-1}(\neg r_{kj} \vee \neg l_{ij}, d_{ij}, k)\Big).
\end{aligned} \tag{25}
$$

B. Multiple PB Constraints

To solve the group MaxSAT formulae $\mathcal{F}'$ in (25), we encode it into a series of SAT instances referring to (10) by simultaneously using multiple PB constraints encoding as follows:

$$
\begin{aligned}
\underset{\text{(potential)}}{\mathcal{F}'_t} &= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n}(\neg x_{ijk} \vee b_{ijk})\\
&\quad \wedge \bigwedge_{k=1}^{m}\underset{\text{PB}}{\mathsf{CNF}}\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij} \cdot b_{ijk}) < t\Big)\\
&= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\underset{\text{PB}}{\mathsf{CNF}}\Big(\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij} \cdot x_{ijk}) < t\Big),\\
\underset{\text{(relative)}}{\mathcal{F}'_t} &= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\bigwedge_{i=1}^{m+n-1}\Big(\bigwedge_{j=1}^{m}(\neg r_{ki} \vee \neg l_{ij} \vee b_{ijk})\\
&\quad \bigwedge_{j=m+1}^{m+n-1}(\neg r_{kj} \vee \neg l_{ij} \vee b_{ijk})\Big)\\
&\quad \wedge \bigwedge_{k=1}^{m}\underset{\text{PB}}{\mathsf{CNF}}\Big(\sum_{i=1}^{m+n-1}\sum_{j=1}^{m+n-1}(d_{ij} \cdot b_{ijk}) < t\Big)\\
&= \mathcal{H} \wedge \bigwedge_{k=1}^{m}\underset{\text{PB}}{\mathsf{CNF}}\Bigg(\sum_{i=1}^{m+n-1}\Big(\sum_{j=1}^{m}(d_{ij} \cdot r_{ki} \cdot l_{ij})\\
&\quad + \sum_{j=m+1}^{m+n-1}(d_{ij} \cdot r_{kj} \cdot l_{ij})\Big) < t\Bigg).
\end{aligned} \tag{26}
$$

Theorem 7. Whether for the potential encodings or the relative encoding, in (26), $\mathcal{F}'_t$ is a CNF that is satisfiable if and only if $\mathcal{F}'$ (in (25)) has a valid assignment whose costs for every group (i.e., $\forall k,\ f(k,\epsilon)$) are less than t at the same time.

Proof. Let's consider that for the potential encodings, the auxiliary variables $\Gamma \rightarrow x_{ijk}$; while for the relative encoding, $\Gamma \rightarrow (r_{ki} \wedge l_{ij})$ if $1 \leq j \leq m$ and $\Gamma \rightarrow (r_{kj} \wedge l_{ij})$ otherwise.

Assume that there exists an assignment $\mathcal{A}$ satisfying $\mathcal{F}'_t$. For any variable b_{ijk} included in soft clauses, if $\mathcal{A}(b_{ijk}) = 1$, then for $\mathcal{F}'$, there exists an assignment $\mathcal{A}'$ consistent with $\mathcal{A}$ such that $\mathcal{A}'(\Gamma) \in \{0,1\}$. In this case, whether $\mathcal{A}'(\Gamma)$ is 0 or 1, $\sum_i\sum_j(d_{ij} \cdot \Gamma) < t$. If $\mathcal{A}(b_{ijk}) = 0$, then for $\mathcal{F}'$, there exists an assignment $\mathcal{A}'$ consistent with $\mathcal{A}$ such that $\mathcal{A}'(\Gamma) = 0$.

Assume that there exists an assignment $\mathcal{A}$ satisfying $\mathcal{F}'$. For any variable Γ included in soft clauses, if $\mathcal{A}(\Gamma) = 1$, then for $\mathcal{F}_t'$, there exists an assignment $\mathcal{A}'$ consistent with $\mathcal{A}$ such that $\mathcal{A}'(\Gamma) = 1$, implying $\mathcal{A}'(b_{ijk}) = 1$. If $\mathcal{A}(\Gamma) = 0$, then for $\mathcal{F}_t'$, there exists an assignment $\mathcal{A}'$ consistent with $\mathcal{A}$ such that $\mathcal{A}'(b_{ijk}) \in \{0, 1\}$, subsuming $\mathcal{A}'(b_{ijk}) = 0$, with the result that $\sum_i \sum_j (d_{ij} \cdot b_{ijk}) < t$. □

Therefore, according to Theorem 7 we can know definitively that the solution of the group MaxSAT formula $\mathcal{F}'$ is equivalent to the solution of the original MMOP in (9).

V. Implementation and Evaluation

We evaluated the existing arithmetic encoding, the proposed encodings, and the IP method experimentally on an Intel Core i7-6850K, 3.6 GHz processor with 32 GB RAM running Ubuntu 18.04. The arithmetic encoding is implemented by using the totalizer encoding of PB constraints [27], [28]. The proposed encodings are two potential encodings based on guiding potential and blocking cycle, respectively, and a relative encoding. In the experiments, we called these comparison methods arithmetic, guide, acyclic, and relative, respectively, for short.

Since there is no open benchmark for the min-max mTSP, we selected several benchmark instances of different sizes from TSPLIB.[2] For each selected instance, we specified different numbers of salesmen and generated their corresponding problem instances through implemented encodings and IP formulations. We used an IP optimizer–CPLEX, version 12.7.1 [29], and a group MaxSAT solver by modifying QMAXSAT [30] to solve all generated problem instances. Each generated problem instance will be solved by the corresponding solver within 3,600 CPU seconds time limit.

In Table I, the notation "insn_m" in the first column indicates that the number of cites is n and the number of salesmen is m, where "ins$n$" corresponds to the name of the original benchmark instance. The cell marked by - indicates that the size of its instance is too large, resulting in a file generated by its encoding being much larger than 10 GB. Thus, it was excluded from our experiment. All methods failed to find the optimal solutions of these instances within the limited time. As the result, for each instance, the number of clauses generated by arithmetic is much larger than that of guide or acyclic, while those of guide and acyclic are approximately the same but also much larger than those of relative. For relatively small-scale instances, in optimized value comparison, acyclic is better than the other encodings, but it is still inferior to ip. However, we found that as the scale of the problem became larger, the performance of relative gradually became better, even the obtained initial solution is not updated until the end, while ip has no initial solution.

In addition, we noticed that for the same original benchmark instance, the smaller ratio of the number of cities to the number of salesmen, the smaller the gap between the group MaxSAT methods and ip. Consequently, we conducted an expanded experiment to increase the number of salesmen of benchmark instance eil76, shown in Table II. We can see that, for the instances with a small ratio of the number of cities to the number of salesmen, relative outperformed ip in the overall comparison of optimized values.

[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/

VI. Conclusion

In this paper, we proposed three CNF encodings for the min-max mTSP. These three encodings are all intended to prevent subtours occurring in the solution of the problem; two of them are based on the vertex potentials, and the third is based on the reachabilities. The correctness of the encodings is rigorously proved as well as the number of generated clauses. We also implemented a group MaxSAT algorithm to solve the encoded min-max mTSP. In terms of the space complexity of the generated problem, our new encodings are significantly improved over the existing encoding. Furthermore, although the proposed approaches are not as effective as the IP method in the performance of small-scale problems, one of them outperforms the IP method for an instance for which the ratio of the number of cities to the number of salesmen is small.

References

[1] B. Brumitt and A. Stentz, "Dynamic mission planning for multiple mobile robots," in *Proceedings of ICRA 1996*. IEEE, 1996, pp. 2396–2401.

[2] C. Okonjo-Adigwe, "An effective method of balancing the workload amongst salesmen," *Omega*, vol. 16, no. 2, pp. 159–163, 1988.

[3] M. Iori and J. Riera-Ledesma, "Exact algorithms for the double vehicle routing problem with multiple stacks," *Comput. Oper. Res.*, vol. 63, pp. 83–101, 2015.

[4] G. Yu, "Min-max optimization of several classical discrete optimization problems," *Journal of Optimization Theory and Applications*, vol. 98, no. 1, pp. 221–242, Jul 1998.

[5] K.-I. Ko and C.-L. Lin, *On the Complexity of Min-Max Optimization Problems and their Approximation*. Springer US, 1995, pp. 219–239.

[6] H. Aissi, C. Bazgan, and D. Vanderpooten, "Complexity of the min-max and min-max regret assignment problems," *Oper. Res. Lett.*, vol. 33, no. 6, pp. 634–640, 2005.

[7] P. M. França, M. Gendreau, G. Laporte, and F. M. Müller, "The *m*-traveling salesman problem with minmax objective," *Transportation Science*, vol. 29, no. 3, pp. 267–275, 1995.

[8] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," *SIAM J. Comput.*, vol. 7, no. 2, pp. 178–193, 1978.

[9] S. Somhom, A. Modares, and T. Enkawa, "Competition-based neural network for the multiple travelling salesmen problem with minmax objective," *Comput. Oper. Res.*, vol. 26, no. 4, pp. 395–407, 1999.

[10] A. Modares, S. Somhom, and T. Enkawa, "A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems," *ITOR*, vol. 6, no. 6, pp. 591–606, 1999.

[11] B. Soylu, "A general variable neighborhood search heuristic for multiple traveling salesmen problem," *Comput. Ind. Eng.*, vol. 90, pp. 390–401, 2015.

[12] R. Necula, M. Breaban, and M. Raschip, "Tackling the bi-criteria facet of multiple traveling salesman problem with ant colony systems," in *Proceedings of ICTAI 2015*, 2015, pp. 873–880.

[13] P. Venkatesh and A. Singh, "Two metaheuristic approaches for the multiple traveling salesperson problem," *Appl. Soft Comput.*, vol. 26, pp. 74–89, 2015.

[14] I. Vandermeulen, R. Groß, and A. Kolling, "Balanced task allocation by partitioning the multiple traveling salesperson problem," in *Proceedings of AAMAS '19*, 2019, pp. 1479–1487.

[15] J. A. Svestka and V. E. Huckfeldt, "Computational experience with an m-salesman traveling salesman algorithm," *Management Science*, vol. 19, no. 7, pp. 790–799, 1973.

[16] B. Gavish, "Note—a note on "the formulation of the m-salesman traveling salesman problem"," *Management Science*, vol. 22, no. 6, pp. 704–705, 1976.

TABLE I
COMPARISON RESULTS BETWEEN VARIOUS METHODS

	arithmetic		guide		acyclic		relative		ip
Instance	#clause	opt. value	#clause	opt. value	#clause	opt. value	#clause	opt. value	opt. value
gr24_2	7.6×10^7	1098	2.8×10^6	862	2.9×10^6	840	2.7×10^4	2609	**770**
gr24_3	8.1×10^7	909	2.8×10^6	767	2.9×10^6	766	3.0×10^4	814	**648**
gr24_4	8.6×10^7	757	2.9×10^6	660	3.0×10^6	674	3.3×10^4	750	**629**
gr24_5	9.1×10^7	697	3.0×10^6	642	3.1×10^6	632	3.6×10^4	676	**594**
bays29_2	2.5×10^8	n.a.y.	7.6×10^6	1495	7.8×10^6	1597	4.8×10^4	2393	**1122**
bays29_4	2.7×10^8	1543	7.8×10^6	1018	8.1×10^6	1008	5.7×10^4	1671	**762**
bays29_6	3.0×10^8	1114	8.3×10^6	807	8.5×10^6	795	6.6×10^4	948	**705**
bays29_8	3.2×10^8	958	8.9×10^6	731	9.1×10^6	697	7.6×10^4	779	**684**
dantzig42_2	-		5.3×10^7	2247*	5.4×10^7	1702*	1.5×10^5	856	**482**
dantzig42_4	-		5.4×10^7	1438	5.5×10^7	614	1.7×10^5	568	**473**
dantzig42_6	-		5.5×10^7	769	5.6×10^7	479	1.8×10^5	490	**395**
dantzig42_8	-		5.7×10^7	506	5.8×10^7	435	2.0×10^5	427	**373**
berlin52_3	-		1.6×10^8	n.a.y.	1.6×10^8	n.a.y.	2.9×10^5	6719	**3754**
berlin52_6	-		1.6×10^8	6879	1.7×10^8	7359	3.4×10^5	4758	**3265**
berlin52_9	-		1.7×10^8	18864*	1.7×10^8	3842	3.8×10^5	3915	**2628**
berlin52_12	-		1.8×10^8	4795	1.8×10^8	3812	4.3×10^5	2936	**2668**
eil76_4	-		-		-		9.3×10^5	438	**333**
eil76_8	-		-		-		1.1×10^6	345	**204**
eil76_12	-		-		-		1.2×10^6	255	**210**
eil76_16	-		-		-		1.3×10^6	**226**	245
rat99_5	-		-		-		2.1×10^6	1864	**1004**
rat99_10	-		-		-		2.3×10^6	1005	**912**
rat99_15	-		-		-		2.6×10^6	862	**773**
rat99_20	-		-		-		2.9×10^6	**826**	n.a.y.
bier127_6	-		-		-		4.4×10^6	139764	**63801**
bier127_12	-		-		-		4.9×10^6	**98107**	n.a.y.
bier127_18	-		-		-		5.5×10^6	**75242**	n.a.y.
bier127_24	-		-		-		6.0×10^6	**70310**	n.a.y.
pr152_8	-		-		-		7.7×10^6	953386*	**105874***
pr152_16	-		-		-		8.7×10^6	**152925**	n.a.y.
pr152_24	-		-		-		9.8×10^6	**129144**	n.a.y.
pr152_32	-		-		-		1.1×10^7	**211209***	n.a.y.

(1) n.a.y.: no answer yet; (2) `value*`: an initial solution, while it cannot be updated until the end of the limit time.

TABLE II
THE EXPANSION COMPARISON OF OPTIMIZED VALUES FOR INSTANCE EIL76

eil76	_18	_20	_22	_24	_26	_28	_30	_32	_34	_36	_38
relative	**214**	**184**	201	**196**	**182**	**184**	**173**	**151**	**138**	168	**176**
ip	247	215	**175**	246	225	265	223	170	210	**141**	179

[17] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.

[18] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of STOC '71*. ACM, 1971, pp. 151–158.

[19] C. M. Li and F. Manyà, "Maxsat, hard and soft constraints," in *Handbook of Satisfiability*, 2009, pp. 613–631.

[20] M. Koshimura, T. Zhang, H. Fujita, and R. Hasegawa, "Qmaxsat: A partial max-sat solver," *JSAT*, vol. 8, no. 1/2, pp. 95–100, 2012.

[21] O. Roussel and V. M. Manquinho, "Pseudo-boolean and cardinality constraints," in *Handbook of Satisfiability*, 2009, pp. 695–733.

[22] C. Ansótegui, M. L. Bonet, and J. Levy, "Sat-based maxsat algorithms," *Artif. Intell.*, vol. 196, pp. 77–105, 2013.

[23] F. Heras, A. Morgado, and J. Marques-Silva, "Maxsat-based encodings for group maxsat," *AI Commun.*, vol. 28, no. 2, pp. 195–214, 2015.

[24] T. Walsh, "SAT v CSP," in *Proceedings of CP 2000*, 2000, pp. 441–456.

[25] T. Paxian, S. Reimer, and B. Becker, "Dynamic polynomial watchdog encoding for solving weighted maxsat," in *Proceedings of SAT 2018*. Springer, 2018, pp. 37–53.

[26] S. D. Prestwich, "SAT problems with chains of dependent variables," *Discret. Appl. Math.*, vol. 130, no. 2, pp. 329–350, 2003.

[27] O. Bailleux and Y. Boufkhad, "Efficient CNF encoding of boolean cardinality constraints," in *Proceedings of CP 2003*, 2003, pp. 108–122.

[28] O. Bailleux, Y. Boufkhad, and O. Roussel, "New encodings of pseudo-boolean constraints into CNF," in *Proceedings of SAT 2009*, 2009, pp. 181–194.

[29] I. I. CPLEX, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[30] A. Zha, M. Koshimura, and H. Fujita, "*N*-level modulo-based CNF encodings of pseudo-boolean constraints for maxsat," *Constraints*, vol. 24, no. 2, pp. 133–161, 2019.