Solving the Employee Task Distribution Problem with Multiple Objectives

Matthias Horn[1], Marie-Louise Lackner[1], Christoph Mrkvicka[2], Nysret Musliu[1], Jakob Preininger[2], and Felix Winter[1]

[1] Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, DBAI, TU Wien, Favoritenstraße 9, 1040 Vienna, Austria
{matthias.horn,marie-louise.lackner,nysret.musliu,felix.winter}@tuwien.ac.at
[2] MCP GmbH, Canovagasse 7, 1010 Vienna, Austria
{christoph.mrkvicka,jakob.preininger}@mcp-alfa.com

Abstract. Assigning employees to operations effectively is a frequent task across diverse industry areas. The challenge of this industry application is to provide a solution method that is flexible enough to be easily adjusted to a specific use case's constraints and optimization objectives. In this paper, we introduce and formally define the Employee Task Distribution Problem (ETDP), describe the highly configurable objective function and propose a solver-independent model that is solved using different constraint programming and mixed integer programming solvers. Furthermore, we prove that the ETDP is NP-hard. We evaluate the performance of our approach using a large benchmark set based on real-life instances and a range of exact state-of-the-art solvers. The best methods can find optimal solutions for nearly all benchmark instances within a realistic time bound for practical usage.

Keywords: Employee Task Distribution Problem, Assignment Problem, Constraint Programming, Integer Programming

1 Introduction

Across diverse industry areas, employees' time capacities must be effectively distributed among operations or tasks on a daily basis. Such an assignment can concern a single or several shifts and must respect various constraints such as capacity limits and employee qualifications. Moreover, a good assignment will make use of the available resources as effectively as possible. What it means to use resources effectively depends on the specific use case; a trade-off between a selection of the following goals needs to be found: maximize the overall output, prioritize bottleneck operations, preferably choose fixed employees instead of temporary or leased staff, maximize the quality of the output, or schedule employees to work on as few different operations as possible within one shift. In addition, when planning several shifts simultaneously, it can be desirable to give employees continuity in their tasks, i.e., to assign employees to the same operations in subsequent shifts.

In this paper, we introduce and define this novel real-life planning problem; we refer to it as the *Employee Task Distribution Problem* (ETDP). This problem originates from collaborations with several industry partners. The challenge of solving this practical problem is twofold: First, the solution method needs to be highly configurable to handle different practical use cases (with varying constraints and objectives). Second, high-quality solutions need to be found quickly to allow for a responsive user experience in industrial settings (the assignments need to be updated often and time efficiently).

The main contributions of this paper are as follows. We propose a solver-independent mathematical model with flexible constraints and an objective function that can handle priorities and relative weights of various objective components. This model is implemented using the high-level modeling language MiniZinc [22]. We evaluate our solution approach in a series of experiments conducted using state-of-the-art *Mixed Integer Programming* (MIP) and *Constraint Programming* (CP) solvers. Our benchmark set consists of 216 instances based on real-life scenarios and is publicly available online [15]. The most successful solvers can find optimal solutions within less than a minute. In addition, we provide computational hardness results for the ETDP, showing that several variants of the ETDP are $\mathcal{NP}$-hard. Thereby, we identify which constraints and components of the objective functions cause computational challenges. These theoretical results are also reflected in our experimental evaluation.

The remainder of the paper is organized as follows: We formally introduce the ETDP and define the objective components in Section 2. Our solver-independent mathematical model is presented in Section 3. We analyze the computational complexity of ETDP in Section 4 and present our experimental evaluation and results in Section 5.

1.1 Related Work

At its core, the ETDP can be seen as a transportation problem for which the employees correspond to suppliers, the operations to customers, and the amount shipped from some supplier to a customer corresponds to the amount of time units the corresponding employee is assigned to the corresponding operation. For a transportation problem, every supplier can ship to any of the customers at a given shipping cost per unit. The goal is to decide on the amounts shipped between every supplier-customer pair to minimize the total cost of meeting customer demands. In the ETDP context, these shipping costs can be interpreted as costs related to employee qualifications, employee types, and operations types. The transportation problem is one of the oldest problems in operations research [20]. See [25,8] for more details on transportation problems. It can be solved efficiently in polynomial time by formulating it as a linear program or minimum-cost flow problem with linear cost function (see the early works [17,14,6,9] and e.g. [26,3,16] for more recent publications). However, due to special constraints and objectives, the ETDP is more complex than the classical transportation problem and these polynomial-time algorithms are not applicable.

The Fixed-Charge Transportation Problem (FCTP) [13,18], is a generalization of the transportation problem for which fixed costs are incurred whenever a route between a supplier and a customer is opened. These fixed costs are added to the unit costs incurred per unit shipped from the given supplier to the given customer. The objective of the FCTP is to decide which routes are opened and which amounts are shipped on these routes to minimize the total cost. This objective is related to the objective of the ETDP to schedule employees to work on as few different operations as possible within one shift. See Theorem 1 in Section 4 for details on the relation of the ETDP to the FCTP.

The ETDP is also related to the assignment problem [24], for which tasks must be assigned to agents to minimize overall costs. In contrast to the ETDP, a one-to-one assignment needs to be found for the assignment problem. Also, the generalized assignment problem (GAP) [4,23] is not general enough to capture the nature of the ETDP. Here, multiple tasks may be assigned to one agent, but tasks may not be split among multiple agents.

Another loosely related problem is the resource-constrained project scheduling problem (RCPSP), which considers scheduling subject to resource and preceding constraints like minimum and maximum time lags. The problem is widely studied in the literature. For an overview see [11,12]. However, the RCPSP focuses much more on the scheduling aspect, whereas our problem abstracts away timing information of operations and employees and focuses more on the assignment part.

2 The Employee Task Distribution Problem

An instance of the *Employee Task Distribution Problem* (ETDP) consists of a set of employees, a set of operations, a set of time buckets, and a set of qualifications. The goal is to assign a number of time units to every bucket-employee-operation triple so that all constraints are respected and the objective function is minimized. Assigning 0 time units to a triple is always admissible.

An overview of all the instance parameters of an ETDP instance is given in Table 1. Let $\mathcal{B} = \{1, 2, \ldots, u\}$ be the set of time buckets. These time buckets are used to model subsequent non-overlapping blocks of time within which the assignments are made. Such time buckets can be interpreted as shifts, work days, calendar weeks, etc. The set of employees $\mathcal{E} = \{1, 2, \ldots, m\}$ is specified by the employees' supplied time capacities $\mathrm{sc}(b, e) \in \mathbb{N} \cup \{0\}\ \forall b \in \mathcal{B}, e \in \mathcal{E}$ and the employees' priority levels $\mathrm{ep}(b, e) \in \mathbb{N}^+$. The set of operations is denoted by $\mathcal{O} = \{1, 2, \ldots, n\}$. For every operation $o \in \mathcal{O}$ and time bucket $b \in \mathcal{B}$, we are given the demanded time capacity $\mathrm{dc}(b, o) \in \mathbb{N} \cup \{0\}$, the priority level $\mathrm{op}(b, o) \in \mathbb{N}^+$, the minimum (positive) assigned capacity $\mathrm{mc}(b, o) \in \mathbb{N} \cup \{0\}$ and $\mathrm{mp}(b, o) \in \mathbb{N} \cup \{0\}$, the maximum number of operations any employee assigned to o can be assigned to in total within the same time bucket. The qualification matrix $Q = (Q(o, e))_{o \in \mathcal{O}, e \in \mathcal{E}}$ with $Q(o, e) \in \mathbb{N} \cup \{0\}$ captures which employees are qualified for which operations. An entry $Q(o, e) = 0$ means that employee e cannot be assigned to operation o and the value of an entry $Q(o, e) > 0$ indicates the level of qualification. Different levels of qualification can be used to model differences in the quality of the output resulting from an assignment: If e_1 and e_2 are to employees that are qualified for an operation $o \in \mathcal{O}$, $Q(o, e_1) > Q(o, e_2) > 0$ means that assigning e_1 to this operation leads to an output of superior quality than assigning e_2 to this operation.

For every bucket-employee-operation triple (b, o, e) with $b \in \mathcal{B}$, $o \in \mathcal{O}$ and $e \in \mathcal{E}$, a non-negative value $A(b, o, e) \in \mathbb{N} \cup \{0\}$ needs to be assigned. This value $A(b, o, e)$ corresponds to the number of time units employee e works on operation o within time bucket e. The constraints that need to be satisfied by the assignments $A(b, o, e)$ for all

Employees $e \in \mathcal{E}$	$\mathrm{sc}(b,e)$ $\mathrm{ep}(b,e)$	supplied capacity employee priority	Constraint (3) Objective (1b)
Operations $o \in \mathcal{O}$	$\mathrm{dc}(b,o)$ $\mathrm{op}(b,o)$ $\mathrm{mc}(b,o)$ $\mathrm{mp}(b,o)$	demanded capacity operation priority min. assigned capacity max. parallel operations	Constraint (3) Objective (1c) Constraint (5) Constraint (6)
Qualifications	$Q(o,e)$	qualification level	Constraint (2) Objective (1d)

Table 1: Overview of the instance parameters of the ETDP for a time bucket $b \in \mathcal{B}$

$b \in \mathcal{B}$, $o \in \mathcal{O}$ and $e \in \mathcal{E}$ as well as the objectives that should be minimized are described in what follows.

2.1 Constraints

A feasible set of assignments $A(b,o,e) \in \mathbb{N} \cup \{0\}$ for all time buckets b, operations o and employees e needs to respect the following constraints. Setting $A(b,o,e) = 0$ is always admissible; a positive assignment $A(b,o,e) > 0$ may however only be made if $Q(o,e) > 0$, i.e., employee e is qualified for operation o. For employees and operations, the supplied resp. demanded capacity levels may not be exceeded for each time bucket. That is, the sum of all assignments to a given operation o (resp. employee e) within a time bucket b may not exceed the demanded capacity $\mathrm{dc}(b,o)$ (resp. supplied capacity $\mathrm{sc}(b,e)$). Moreover, for operations with minimum assigned capacity $\mathrm{mc}(b,o) > 0$, the sum of all assignments must be equal to 0 or at least equal to $\mathrm{mc}(b,o)$. The maximum-parallel-operations constraint enforces that any employee e assigned to an operation o at time bucket b with $\mathrm{mp}(b,o) > 0$ may not be assigned to more than $\mathrm{mp}(b,o)$ many operations in total. Both the minimum-assigned-capacity and the maximum-parallel-operations constraints alone cause the ETDP to be $\mathcal{NP}$-hard. For a formal statement and proof of this result, see Section 4.

2.2 Objectives

A multitude of different practical use cases give rise to different optimization objectives for the ETDP. These use cases can occur independently of each other within a specific application. Often, a trade-off between several, potentially conflicting, objectives needs to be found in practice. Finding such a trade-off among is the topic of multiobjective optimization and many different methods have been suggested to approach this goal [7,19]. For our specific application, we wanted an approach that allows high flexibility and allows the users to select a combination of objectives, each with a certain priority and weight. Priorities can be used to set the order of lexicographic optimization and weights can additionally be used to balance objectives with the same priority level.

Basic objective: Maximize the sum of assignments The basic objective of the ETDP is to maximize the overall satisfied capacity demands over all time buckets. Equivalently, the objective is to minimize the amount of unsatisfied capacity demands, i.e., the cost is defined as the difference between the theoretical maximum assignment level and the sum of all assignments.

Employee prioritization If capacity supplies exceed capacity demands, i.e., not all capacity supplies are fully used, the distribution of used capacity per employee should be done so that permanent staff are lexicographically more important than others, e.g. leased or temporary staff. With the usage of $\mathrm{ep}(b, e) \in \mathbb{N}$, many levels of employee prioritization are possible.

Operation prioritization If capacity demands cannot be met fully, the distribution of assigned capacity per operation should be such that bottleneck operations are lexicographically more important than others. Levels of operation prioritization are specified with $\mathrm{op}(b, o) \in \mathbb{N}$ for $o \in O$.

Maximize qualification score The quality of the output might depend on the qualification level of employees assigned to operations. In this case, the goal is to maximize the weighted sum of assignments, where the qualification levels give the weights. Note that this objective is equivalent to the basic objective if the qualification matrix Q is a binary matrix. Qualification levels can also be used to model employees' preferences for certain operations. In this case, this objective corresponds to maximizing employees' satisfaction with the assignments.

Time bucket change objective If multiple time buckets are considered, another criterion for the output's quality is the degree of continuity in the tasks assigned to employees. More precisely, if employee e is assigned to operation o in time bucket b, i.e. $A(b, o, e) > 0$, it can be beneficial to assign this employee to the same operation in the consecutive time bucket $b+1$, i.e. to set $A(b+1, o, e) > 0$ as well. Maximizing the overall continuity across all employees is achieved by minimizing the number of times this is not the case for every employee, i.e. minimizing the number of operations and time buckets for which $A(b, o, e) > 0$ but $A(b+1, o, e) = 0$.

Concentrated assignments = minimize assignment count objective In practice, solutions in which the assignments are concentrated are often more favorable than those with assignments spread across the operations and employees. It is better for the employees and the outcome of their work if they are assigned to fewer operations within the same shift. For instance, a solution in which $A(b_1, o_1, e_1) = 2$ and $A(b_1, o_2, e_2) = 2$ is often clearly better than a solution with $A(b_1, o_1, e_1) = A(b_1, o_1, e_2) = A(b_1, o_2, e_1) = A(b_1, o_2, e_2) = 1$ even if all other objective values are equal.

The sole goal of minimizing the number of assignments would always lead to a null assignment. Therefore, it only makes sense in combination with one of the other objectives, e.g. minimizing the number of assignments while at the same time maximizing the sum of assignments. While all other objectives are linear in the assignment values $A(b, o, e)$, the "minimize-assignment-count" and the "time-bucket-change"-objective are not. Using the "minimize-assignment-count" objective combined with any of the other ones causes the ETDP to be $\mathcal{NP}$-hard, as we show in Section 4.

See the mathematical model introduced in the following section for a formal definition of these objective functions and an explanation of how they are combined into a single function using weights. Note that we formulate all objective functions as cost functions, i.e., we formulate the ETDP as a minimization problem.

3 Mathematical Model

We use decision variables

$$A(b,o,e) \in \mathbb{N} \cup \{0\} \quad \forall b \in \mathcal{B} \quad \forall o \in \mathcal{O} \quad \forall e \in \mathcal{E}$$

to indicate the amount of capacity assigned to a bucket, operation, and employee triple. In the following, we have underlined all constants involved. For their calculation, we refer the reader to Section 3.3 and the technical appendix[15].

$$\text{Min.} \quad obj = \underline{\lambda_a} \cdot a + \underline{\lambda_e} \cdot e + \underline{\lambda_o} \cdot o + \underline{\lambda_q} \cdot q + \underline{\lambda_r} \cdot r + \underline{\lambda_c} \cdot c, \text{ where} \tag{1}$$

$$a = \underline{max_a} - \sum_{b\in\mathcal{B}, o\in\mathcal{O}, e\in\mathcal{E}} A(b,o,e) \tag{1a}$$

$$e = \underline{max_e} - \sum_{\substack{p\in\{1,\ldots,p_E\},\\ b\in\mathcal{B}}} \underline{\text{bigM}^{\text{e}}(b,p)} \left(\sum_{\substack{e\in\mathcal{E}:\,\text{ep}(b,e)=p\\ o\in\mathcal{O}}} A(b,o,e) \right) \tag{1b}$$

$$o = \underline{max_o} - \sum_{\substack{p\in\{1,\ldots,p_O\},\\ b\in\mathcal{B}}} \underline{\text{bigM}^{\text{o}}(b,p)} \left(\sum_{\substack{o\in\mathcal{O}:\,\text{op}(b,o)=p\\ e\in\mathcal{E}}} A(b,o,e) \right) \tag{1c}$$

$$q = \underline{max_q} - \sum_{b\in\mathcal{B}, o\in\mathcal{O}, e\in\mathcal{E}} (A(b,o,e) \cdot Q(o,e)) \tag{1d}$$

$$r = \sum_{\substack{b\in\{1,\ldots,u-1\},\\ o\in\mathcal{O},\\ e\in\mathcal{E}}} |\mathbf{1}_{>0}(A(b,o,e)) - \mathbf{1}_{>0}(A(b+1,o,e))| \tag{1e}$$

$$c = |\{b \in \mathcal{B}, o \in \mathcal{O}, e \in \mathcal{E} : A(b,o,e) > 0\}| \tag{1f}$$

$$\text{s.t. } A(b,o,e) \leq Q(o,e) \cdot \underline{cap} \qquad \forall b \in \mathcal{B}, \forall o \in \mathcal{O}, \forall e \in \mathcal{E} \tag{2}$$

$$\sum_{o\in\mathcal{O}} A(b,o,e) \leq \text{sc}(b,e) \qquad \forall b \in \mathcal{B}, \forall e \in \mathcal{E} \tag{3}$$

$$\sum_{e\in\mathcal{E}} A(b,o,e) \leq \text{dc}(b,o) \qquad \forall b \in \mathcal{B}, \forall o \in \mathcal{O} \tag{4}$$

$$\sum_{e\in\mathcal{E}} A(b,o,e) = 0 \vee \sum_{e\in\mathcal{E}} A(b,o,e) \geq \text{mc}(b,o) \qquad \forall b \in \mathcal{B}, \forall o \in \mathcal{O} \tag{5}$$

$$\begin{aligned} &\forall b \in \mathcal{B}, \forall o \in \mathcal{O} \text{ with } \text{mp}(b,o) > 0,\\ &\forall e \in \mathcal{E} \text{ with } A(b,o,e) > 0 :\\ &|\{i \in \mathcal{O} : A(b,i,e) > 0\}| \leq \text{mp}(b,o) \end{aligned} \tag{6}$$

3.1 Objective function

The objective function in equation (1) consists of six components combined in a weighted sum. These weights $\underline{\lambda_i}$ are calculated on the basis of the priorities and weights of every objective component and are calculated as "big M" values. The use of large constants that are referred to as "big Ms" is very common when solving multi-objective optimization problems, particularly in practical applications. Note that all solutions that are optimal with respect to a weighted sum of the objective components are Pareto-optimal [5] with respect to the three objective components. All objective components are modeled as cost functions, i.e., we formulate the ETDP as a minimization problem. The objective component a corresponds to the basic objective, e to employee prioritization, o to operation prioritization, q to maximizing the qualification score, r the time bucket change objective with indicator function $\mathbf{1}_{>0}(x)$, which is one if $x > 0$ and zero otherwise, and c is the number of non-zero assignments. Note that all objective functions are linear in the decision variables except for r and c. The constants $\underline{max_a}$, $\underline{max_e}$, $\underline{max_o}$ and $\underline{max_q}$ are chosen in such a way for every instance that a solution with cost value equal to 0 corresponds to a theoretical optimum. We refer the reader to the online appendix for detailed calculations [15].

3.2 Constraints

The constraint in equation (2) ensures that an assignment of employee e to operation o at time bucket b, i.e. $A(b, o, e) > 0$, is only possible if employee e is qualified for operation o, i.e. $Q(o, e) > 0$. The constant $\underline{cap}$ is the minimum of the overall maximum supplied capacity and the overall maximum demanded capacity. Equations (3) and (4) encode the capacity constraints of employees and operations. Equation (5) encodes the minimum-assigned-capacity constraint: The sum of all assignments to operation o needs to be equal to 0 or greater than or equal to $\mathrm{mc}(b, o)$ for bucket b. This constraint is thus always fulfilled for operations with $\mathrm{mc}(b, o) = 0$. Finally, the maximum-parallel-operations constraint is modeled by Equation (6) and has to be fulfilled for all operations $o \in \mathcal{O}$ and buckets $b \in \mathcal{B}$ with $\mathrm{mp}(b, o) > 0$: Every employee e assigned to such an operation can in total be assigned to no more than $\mathrm{mp}(b, o)$ many different operations.

3.3 Weights of the objective components

The weights of objective components $\underline{\lambda_a}$, $\underline{\lambda_e}$, $\underline{\lambda_o}$, $\underline{\lambda_q}$, $\underline{\lambda_r}$ and $\underline{\lambda_c}$ are set in order to respect the priorities and weights of objective components in the aggregated objective function obj as defined in equation (1).

For this purpose, we introduce the following notation. The objectives a, e, o, q, r and c are first grouped into objectives with same priority level p. Note that we assume that the objective priorities take all values between 1 and the maximum objective priority k. Within a priority level p, the aggregated objective function obj_p is calculated as a weighted sum of the normalized objective components $obj_{p,1}, \ldots, obj_{p,k_p}$ using the respective weights $w_{p,1}, \ldots, w_{p,k_p}$, where k_p is the number of objective functions with priority level p. The overall aggregated objective function obj is then created as a linear combination of the functions $obj_1, \ldots, obj_k$ where the coefficients in this sum

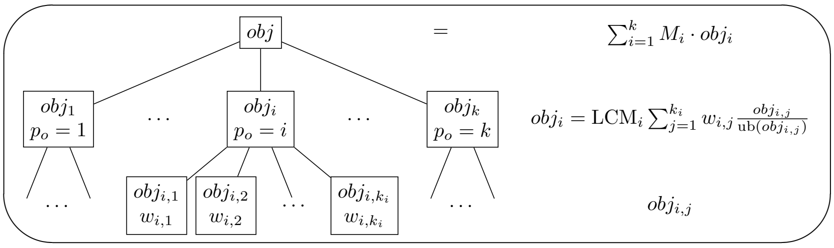

Fig. 1: Aggregated objective function with weights and priorities

are chosen as big M constants such that a lexicographic optimization is achieved (obj_1 is lexicographically more important than obj_2, aso.). For a visualization of the structure of the aggregated objective function, see Figure 1.

The objective function obj is defined as follows:

$$
\begin{aligned}
obj &= \sum_{i=1}^{k} M_i \cdot obj_i \qquad \text{with} \\
M_i &= \prod_{j=i+1}^{k} (1 + \mathrm{ub}(obj_i)) \\
&= \prod_{j=i+1}^{k} \left(1 + \mathrm{LCM}_i \cdot \sum_{j=1}^{k_i} w_{i,j}\right) \\
obj_i &= \mathrm{LCM}_i \sum_{j=1}^{k_i} w_{i,j} \frac{obj_{i,j}}{\mathrm{ub}(obj_{i,j})} \leq \mathrm{LCM}_i \cdot \sum_{j=1}^{k_i} w_{i,j} \\
\mathrm{LCM}_i &= \frac{\mathrm{LCM}\left(\mathrm{ub}(obj_{i,j}) : j \in [1, k_i]\right)}{\mathrm{GCD}\left(w_{i,j} : j \in [1, k_i]\right)}
\end{aligned}
$$

The *least common multiple* LCM and the *greatest common divisor* GCD are used to ensure that the objective function is an integer to make it applicable for constraint programming solvers. The functions $\mathrm{ub}(obj_{i,j})$ are upper bounds on the respective objectives $obj_{i,j}$ which can be one of the cost functions a, e, o, q, r, and c. Their values are given by the constants defined in the previous section:

$$
\begin{aligned}
\mathrm{ub}(a) &= \underline{max_a} \text{ as in equation (1a)} \\
\mathrm{ub}(e) &= \underline{max_e} \text{ as in equation (1b)} \\
\mathrm{ub}(o) &= \underline{max_o} \text{ as in equation (1c)} \\
\mathrm{ub}(q) &= \underline{max_q} \text{ as in equation (1d)} \\
\mathrm{ub}(r) &= |\{o \in \mathcal{O}, e \in \mathcal{E} : Q(o, e) > 0\}| \cdot (u - 1). \\
\mathrm{ub}(c) &= |\{o \in \mathcal{O}, e \in \mathcal{E} : Q(o, e) > 0\}|.
\end{aligned}
$$

We can also write the objective function as follows:

$$obj = \sum_{f \in \mathcal{F}} \underline{\lambda_f} \cdot f = \sum_{f \in \mathcal{F}} \left(M_{p_f} \cdot \frac{\mathrm{LCM}_{p_f}}{\mathrm{ub}(f)} \cdot w_f \cdot f \right),$$

where p_f is the priority and w_f is the weight of objective function (name) f from the set of objective functions $\mathcal{F} = \{a, e, o, q, r, c\}$. The weights $\underline{\lambda_f}$ for $f \in \mathcal{F}$ are the weights used in the mathematical model.

4 Complexity Results

The Employee Task Distribution Problem with the basic objective (1a), the employee prioritization (1b), operation prioritization (1c) or qualification score objective (1d) but without max-parallel-operations-constraints (6) and without min-assigned-capacity-constraints (5) can be formulated as a classical transportation problem [25,8]. These special cases of the ETDP can thus also be solved in polynomial time.

However, very simplified versions of the ETDP involving the other objectives and the specialized max-parallel-operations- and min-assigned-capacity-constraints are $\mathcal{NP}$-hard. In the following, we formally state these $\mathcal{NP}$-hardness results for the ETDP. Note that some proofs can be found in the technical appendix, which is available online [15].

Theorem 1. *The Employee Assignment optimization problem without max-parallel-operations- and min-assigned-capacity-constraints and with the objective of minimizing the number of assignments as defined in equation* (1f) *and the quality score objective (equation* (1d)*) or sum of assignments (equation* (1a)*) is strongly $\mathcal{NP}$-hard.*

Proof. The ETDP without max-parallel-operations- and min-assigned-capacity-constraints and with the objective of minimizing the number of assignments and the quality score objective (or sum of assignments) is equivalent to a special case of the *Fixed-Charge Transportation Problem* (FCTP). The FCTP is a generalization of the transportation problem and was introduced by Hirsch and Dantzig [13]. The following description of the FCTP is given by Kowalski [18]:

The fixed-charge transportation problem consists of m suppliers and n customers. Each of the m suppliers can ship to any of the n customers at a shipping cost per unit $c_{i,j}$ (unit cost for shipping from supplier i to customer j), plus a fixed cost $f_{i,j}$, assumed for opening this route. The objective is then to determine which routes are to be opened and the size of the shipment on those routes so that the total cost of meeting demand, given the supply constraints, is minimized.

The FCTP can be formalized as follows, where the decision variables $x_{i,j}$ model the amount shipped from supplier i to customer j and the $\{0, 1\}$-variables $y_{i,j}$ model whether a positive amount is shipped from supplier i to customer j or not:

$$\text{minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{i,j} x_{i,j} + f_{i,j} y_{i,j})$$

$$\text{subject to } \sum_{j=1}^{n} x_{i,j} = a_i$$
$$\sum_{i=1}^{m} x_{i,j} = b_j$$
$$x_{i,j} \geq 0 \quad \forall i, j$$
$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_i$$
$$y_{i,j} = 0 \text{ if } x_{i,j} = 0$$
$$y_{i,j} = 1 \text{ if } x_{i,j} > 0$$

Note that setting $f_{i,j}$ to some fixed weight corresponding to the weight w_c of the objective "minimizing the number of assignments" allows us to model the ETDP (without max-parallel-operations and min-assigned-capacity constraints) as FCTP. The costs $c_{i,j}$ need to be set accordingly to model the qualification levels, setting them to large "big M" values in case of absent qualifications. Moreover, dummy operations or tasks must be introduced if the sum of demanded capacities is not equal to the sum of supplied capacities.

Angulo et al. [1] showed that the Fixed Charge Transportation Problem is strongly $\mathcal{NP}$-hard by a reduction from the strongly $\mathcal{NP}$-complete 3-Partition problem. The $\mathcal{NP}$-hardness of the problem holds even if the fixed costs $f_{i,j}$ are constant across all suppliers i and customers j. This shows the $\mathcal{NP}$-hardness of this version of the ETDP.

Theorem 2. *The Employee Assignment optimization problem with min-assigned-capacity-constraint and the sole objective of maximizing the sum of assignments is strongly $\mathcal{NP}$-hard.*

Proof. We prove the $\mathcal{NP}$-hardness of the ETDP with minimum-assigned-capacity-per-operation-Constraint by providing a polynomial time reduction from the $\mathcal{NP}$-hard 3-dimensional Matching problem [10]. See the technical appendix [15] for the complete proof.

Theorem 3. *The Employee Assignment optimization problem with maximum-parallel-operations-constraint and the sole objective of maximizing the sum of assignments is strongly $\mathcal{NP}$-hard.*

Proof. We prove the $\mathcal{NP}$-hardness of the ETDP with maximum-parallel-operations-constraint by providing a polynomial time reduction from the strongly $\mathcal{NP}$-complete 3-Partition problem [10]. See the technical appendix [15] for the complete proof.

Note however that if the max-parallel-operations-constraint is set to $\text{mp}(o) = 1$ for all operations $o \in \mathcal{O}$, the ETDP corresponds to a Single-Source Transportation Problem [21], which is equivalent to a Generalized Assignment Problem [4,23] and is solvable in polynomial time.

5 Experimental Evaluation

We implemented the mathematical model presented earlier using the high-level constraint modeling language MiniZinc [22]. In order to evaluate the performance of our model, we conducted experiments with the following state-of-the-art MIP and CP solvers: Gurobi, CPLEX, Cbc, Chuffed and OR-Tools. For the MIP solvers, MiniZinc automatically converts the given constraint model into an MIP model [2].

The benchmark set used for these experiments consists of a total of 216 instances and is publicly available online [15]. It is based on twelve relatively small real-life instances that were provided to us by our industrial partner. These instances have between 9 and 23 employees and between 5 and 16 operations; at most 448 capacity units need to be distributed among the employees and operations. For every one of these instances, six different settings for the objective function were evaluated in order to model different practical use cases:

- UCOpPrio: the sole objective is operation prioritization
- UCEmpPrio: the sole objective is employee prioritization
- UCAss: the objective is to maximize the sum of assignments with first priority and to minimize the assignment count with second priority
- UCQuali: the sole objective is to maximize the qualification score
- UC3Buckets: three time buckets are considered, a combined objective with the first priority being the maximization of the qualification score and the second priority being the minimization of changes between time buckets with weight 1 and the minimization of the assignment count with weight 2
- UC4Buckets: four time buckets are considered with the same combined objective as for the UC3Buckets use case

In order to test the scalability of our model, we created larger instances from real-life instances by copying the employees and operations and randomly perturbing their properties such as capacities and constraints. Qualifications were also perturbed. This resulted in a set of 72 medium-sized instances for which the number of employees and operations from the real-life instances was doubled and a set of 72 large instances for which they were multiplied by five. The large instances thus have up to 115 employees and up to 80 operations.

In practice, high-quality solutions to the ETDP need to be found very quickly. Indeed, even though the employee-operation-assignments are planned in advance over a large time horizon of several weeks or months, the actual capacities of employees and operations are subject to short-time changes due to sickness, material shortage, order

	feasible solutions found (in %)				opt. solved	proven opt.	opt. gap (in %)	
solver	total	small	medium	large	(in %)	(in %)	average	std
Gurobi	**100.0**	**100.0**	**100.0**	**100.0**	**84.7**	**84.7**	**2.1**	**12.0**
CPLEX	**100.0**	**100.0**	**100.0**	**100.0**	83.3	79.6	2.8	14.2
Cbc	97.2	**100.0**	**100.0**	91.7	65.7	64.4	11.1	29.2
Chuffed	86.6	98.6	88.9	72.2	10.6	6.9	68.5	39.2
OR-Tools	**100.0**	**100.0**	**100.0**	**100.0**	3.2	1.9	79.2	31.2

Table 2: Overview of the experimental results achieved with a time limit of 60 seconds.

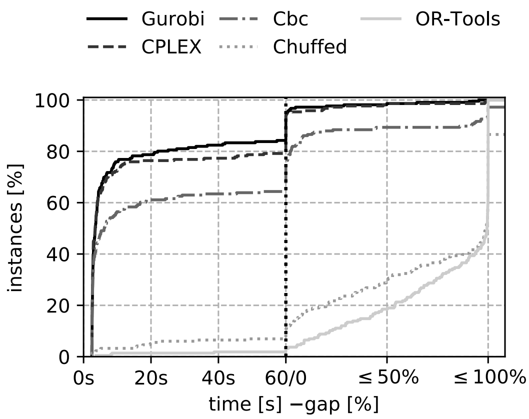

Fig. 2: Performance plot over all 216 instances, comparing solvers Gurobi, CPLEX, Cbc, Chuffed, and OR-Tools.

cancellations etc. Therefore, (updated) instances of the ETDP need to be solved often and quickly. We thus conducted experiments with a runtime limit of 60 seconds, which is a realistic time-bound for practical purposes. We also experimented with a 3-minute time limit but do not report these results here in detail as these only revealed minor qualitative differences in the solvers' performance. All experiments were run on single cores, using a computing cluster with ten identical nodes, each having 24 cores, an Intel(R) Xeon(R) CPU E5–2650 v4 @ 2.20GHz and 252 GB RAM.

5.1 Experimental Results

An overview of our experimental results with a runtime limit of 60 seconds can be found in Table 2. For all solvers, we display the percentages of instances for which (i) a feasible solution (in total, for small, medium and large instances), (ii) an optimal solution, and (iii) an optimal solution, including an optimality proof, could be found within the time limit. Note that the difference between (ii) and (iii) is that (iii) reports the percentage of instances that the corresponding approach could solve to proven optimality, whereas (ii) also includes instances where the approach was able to find the optimal solution, but not necessarily with an optimality proof (but we know that the solution is optimal due to a lower bound obtained from one of the other approaches). Moreover, the average optimality gaps (in percentage) with the corresponding standard deviations are reported. The optimality gap for a given instance i is defined as follows: $g(i) = (c(i) - b(i))/c(i) \cdot 100$, where $c(i)$ is the cost of the found solution and $b(i)$ is the best lower bound on the solution cost found by any of the evaluated solvers. In this table, the best results are highlighted in bold font.

A first observation is that some solvers were not capable of finding feasible solutions for all 216 benchmark instances, even though the trivial null assignment—assigning 0

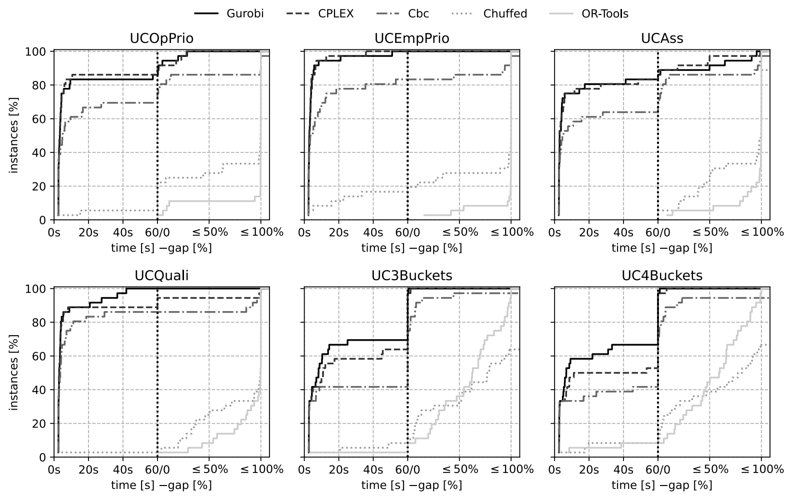

Fig. 3: Performance plot over all 216 instances grouped by the use cases. Use cases with a linear objective function can be solved more efficiently than use cases with non-linear objectives.

to every bucket-operation-employee-triple—is always a feasible solution. Indeed, only Gurobi, CPLEX, and OR-Tools found solutions for all instances, but Cbc failed to find a feasible solution for some large instances. Chuffed was overall less successful at finding solutions. Almost all solutions found by Gurobi were of very high quality, as optimality proofs could be delivered for more than 84 % of all instances. Moreover, for those instances that were not provably solved to optimality within the time limit of 60 seconds, the solutions found were also close to the optimum, as the average optimality gap is 2.1 %.

In terms of optimality, the second-best results could be achieved by CPLEX (roughly 83 % of optimally solved instances), followed by Cbc (slightly less than 66 % of optimally solved instances). Both Gurobi and CPLEX were capable of finding optimal solutions for almost all real-life instances and all use cases. Chuffed and OR-Tools only managed to solve very few instances to optimality and are clearly not competitive with the MIP solvers Gurobi, CPLEX and Cbc for this model. For all solvers except Gurobi and CPLEX, the optimality gaps get large, with average values between 11 and 80 %. Note that for those instances where some solver could find no feasible solution, we used the solution cost of the null assignment to compute the optimality gap.

These results are also presented graphically as a performance plot in Figure 2. The plot is divided into two parts: The left part shows the number of instances solved to proven optimality within a certain number of seconds. For instances that could not be solved to proven optimality within 60 seconds, the right part shows the remaining optimality gap for the number of instances.

Figure 2 also indicates that almost all optimality proofs are delivered by Gurobi, CPLEX and Cbc within the first 20 seconds. More precisely, Gurobi, CPLEX and Cbc were able to prove optimality of 93 %, 96 %, and 95 % of the optimality solved instances, respectively. Moreover, the median proof times of the instances solved to proven optimality are 2.95, 2.83, and 2.86 seconds, for Gurobi, CPLEX and Cbc, respectively. The slowest median proof times are obtained from Chuffed with 15.14 seconds, followed by OR-Tools with 8.24 seconds.

We also conducted experiments with a runtime limit of five minutes. Increasing the time limit allowed the weaker solvers, Chuffed and OR-Tools, to solve more instances and all solvers to deliver some more optimality proofs.

Moreover, we evaluated the results for the six different use cases separately. The results are shown in Figure 3 where for each use case a performance plot is drawn. All solvers could achieve similar results for the use cases UCOpPrio, UCEmpPrio, and UCQuali, with the highest number of optimally solved instances for these three use cases. The other three use cases UCAss, UC3Buckets, and UC4Buckets were harder to solve. This can be explained as follows: UCOpPrio, UCEmpPrio and UCQuali consist of a single, linear objective function that can be handled efficiently by the evaluated solvers. The other use cases were harder to solve. For the other three use cases UCAss, UC3Buckets, and UC5Buckets, Gurobi delivered the overall best results but could not provide optimality proofs within the time limit for 17 %, 28 %, and 33 % of the instances, respectively. Use case UCAss has a non-linear component (counting the number of positive assignments) and use cases UC3Buckets and UC4Buckets consist of a combination of several objectives with two non-linear components (counting the number of positive assignments and the number of employee-operation changes between time buckets), which slows down the solution process. These experimental results also reflect the complexity results from Section 4. Indeed, the objective functions for use cases UCOpPrio, UCEmpPrio and UCQuali can also be expressed as cost functions in a classical transportation problem, which can be solved in polynomial time. However, computational complexity remains also for these use cases due to the presence of max-parallel-operations- and min-assigned-capacity-constraints. Moreover, the use case UCAss corresponds to an $\mathcal{NP}$-hard variant of the ETDP.

6 Conclusion

In this paper, we introduced and formally defined the Employee Task Distribution Problem, a planning problem that arises in practice and has similarities with classical transportation problems. In order to handle a variety of different objective functions and specialized constraints that distinguish the ETDP from transportation problems, we propose a new solver-independent mathematical model. Our experiments conducted with five state-of-the-art solvers on a benchmark set of 216 instances showed that overall best results could be achieved using the MIP solver Gurobi which could provide optimality proofs within less than 60 seconds for more than 84 % of all instances. The MIP solvers CPLEX and Cbc could also achieve high-quality results. All real-life instances provided by our industry partner could be solved optimally within this very short time bound by Gurobi and CPLEX. Overall, the MIP solvers found better solutions for the large majority of instances in our experiments compared to the evaluated CP solvers, indicating that exploiting the linear structure that lies at the core of the problem is crucial for solving the ETDP efficiently.

As a next step, we plan to develop metaheuristic solution methods for the ETDP independent of external solvers. The challenge is to design algorithms that are capable of finding near-optimal solutions within a few seconds. This would significantly increase the practical applicability of our approach. Developing such solutions would allow our industrial partner to include them in a cloud-based web service and offer them to an even more diverse set of customers.

Acknowledgements The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

1. Angulo, G., Van Vyve, M.: Fixed-charge transportation problems on trees. Operations Research Letters **45**(3), 275–281 (2017)
2. Belov, G., Stuckey, P.J., Tack, G., Wallace, M.: Improved linearization of constraint programming models. In: CP. Lecture Notes in Computer Science, vol. 9892, pp. 49–65. Springer (2016)
3. Bertsekas, D.P., Castanon, D.A.: The auction algorithm for the transportation problem. Annals of Operations Research **20**(1), 67–96 (1989)
4. Cattrysse, D.G., Van Wassenhove, L.N.: A survey of algorithms for the generalized assignment problem. European journal of operational research **60**(3), 260–272 (1992)
5. Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F.: Comparison of multi-objective optimization methodologies for engineering applications. Computers & Mathematics with Applications **63**(5), 912–942 (2012)
6. Dantzig, G.B.: Application of the simplex method to a transportation problem. Chapter XXIII in Activity analysis and production and allocation, Cowles Commission Monograph No. 13, (ed.) T. C. Koopmans (1951)
7. Deb, K.: Multi-objective optimization. In: Search methodologies, pp. 403–449. Springer, Boston, MA (2014)
8. Díaz-Parra, O., Ruiz-Vanoye, J.A., Bernábe Loranca, B., Fuentes-Penna, A., Barrera-Cámara, R.A.: A survey of transportation problems. Journal of Applied Mathematics (2014)
9. Ford Jr, L.R., Fulkerson, D.R.: Solving the transportation problem. Management Science **3**(1), 24–32 (1956)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
11. Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research **207**(1), 1–14 (2010)
12. Hartmann, S., Briskorn, D.: An updated survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research **297**(1), 1–14 (2022)
13. Hirsch, W.M., Dantzig, G.B.: The fixed charge problem. Naval Research Logistics Quarterly **15**(3), 413–424 (1968)
14. Hitchcock, F.L.: The distribution of a product from several sources to numerous localities. Journal of mathematics and physics **20**(1-4), 224–230 (1941)

15. Horn, M., Lackner, M.L., Mrkvicka, C., Musliu, N., Walkiewicz, D., Winter, F.: Benchmark instances, models and technical appendix for the Employee Task Distribution Problem [Data Set] (Dec 2023), https://owncloud.tuwien.ac.at/index.php/s/JltgU8OiozNaTKf
16. Juman, Z., Hoque, M.: An efficient heuristic to obtain a better initial feasible solution to the transportation problem. Applied Soft Computing **34**, 813–826 (2015)
17. Kantorovich, L.V.: Mathematical methods of organizing and planning production. Publication House of the Leningrad State University (1939). Translated in Management science **6**(4), 366–422 (1960)
18. Kowalski, K.: On the structure of the fixed charge transportation problem. International journal of mathematical education in science and technology **36**(8), 879–888 (2005)
19. Miettinen, K.: Nonlinear multiobjective optimization, vol. 12. Springer Science & Business Media, New York (2012)
20. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année pp. 666–704 (1781)
21. Nagelhout, R.V., Thompson, G.L.: A single source transportation algorithm. Computers & Operations Research **7**(3), 185–198 (1980)
22. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: Towards a Standard CP Modelling Language. In: Bessière, C. (ed.) Principles and Practice of Constraint Programming – CP 2007. pp. 529–543. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2007)
23. Öncan, T.: A survey of the generalized assignment problem and its applications. INFOR: Information Systems and Operational Research **45**(3), 123–141 (2007)
24. Pentico, D.W.: Assignment problems: A golden anniversary survey. European Journal of Operational Research **176**(2), 774–793 (2007)
25. Raju, N.: Transportation problem. In: Operations Research: Theory and Practice, chap. 5, pp. 193–290. CRC Press (2019)
26. Vasko, F.J., Storozhyshina, N.: Balancing a transportation problem: Is it really that simple? OR insight **24**(3), 205–214 (2011)