MISTA 2015

Fairness in employee scheduling

Erica Stockwell-Alpert · Christine Chung

1 Introduction

In commercial, industrial, and retail settings, it can be tedious and difficult to find a schedule for workers that fills every shift, gives every employee the hours they need, and does not exceed the company's budget. There may be a large number of shifts of varying lengths, all of which require a minimum amount of coverage, as well as a large number of employees who each require a different number of hours of work per week (or scheduling period).

We consider the problem of finding a work schedule that satisfies all employee and shift requirements: where each employee has a minimum number of hours they must work, each shift must be covered by a minimum number of employees, each employee has a set of shifts they are available to work, and there is a limit on the total number of hours available for distribution. We show that deciding whether a feasible schedule exists (one where each employee is working at least their required number of hours) is NP-complete. We then consider the corresponding NP-hard optimization goal of finding the most fair schedule, where the least "satisfied" employee is as satisfied as possible. That is, we wish to minimize the maximum $r_i - h_i$, where h_i is the hours assigned to employee i and r_i is the number of hours employee i is required to work.

In game-theoretic terms, rather than the *utilitarian* objective of maximizing the average satisfaction over the employees, we focus on the *egalitarian* objective of maximizing the satisfaction of the least-satisfied employee, i.e., looking for a solution that is as *fair* as possible.

We are intentionally ambiguous about the source of the parameter r_i, since it can be interpreted as the number of hours the employee specifies they wish to work, or might instead be specified by an employer/manager as a minimum number of hours an employee must work based on work regulations, bookkeeping logistics, etc.

Our problem, which we call the Employee Satisfaction Problem (ESP), has similarities to many employee timetabling problems (ETPs) that have been studied. Much of the work on

Erica Stockwell-Alpert
NorthPoint Digital, Boston, MA
E-mail: estockwell-alpert@northpointdigital.com

Christine Chung
Department of Computer Science, Connecticut College, New London, CT
E-mail: cchung@conncoll.edu

ETPs has been in the artificial intelligence and operations research communities; numerous experimental studies have been conducted using heuristic methods such as local search, branch-and-bound, genetic algorithms, constraint programming and ILP solvers, e.g., see [1–3, 10, 18, 20, 21]. However the variants of timetabling problems in previous works have different constraints and objectives from ours. In particular, they do not focus on fairness to employees. Another point of contrast is that we provide a formal worst-case guarantee on the quality of our algorithm's solution relative to the optimal solution.

Another important distinction between our problem and many other timetabling problems is that our employee requirements are defined in terms of hours, rather than shifts. The difference in the shift lengths is a very real and practical issue, e.g., an employee who is given a series of 3- or 4-hour shifts rather than 8-hour shifts may not actually be working enough hours to support themselves. And indeed, the differing shift durations are the crucial factor in the intractability of ESP.

1.1 Related work

There is a prodigious amount of work done in timetabling, shift assignment, personnel scheduling, and the like. We mention here some of the works that have more significant similarities to ours.

The ESP is similar in nature to the timetable problem studied early on by Gotlieb [16, 15], which Even et al. later show to be NP-complete [13]. They consider the problem of scheduling teachers in a school to class periods, and their problem has many parallels to ours. But while classes may only be taught by one teacher, the work shifts ("classes") in our problem each have a positive integer parameter specifying the minimum number of employees ("teachers") that must be assigned to it. Each teacher in Gotlieb's problem also has a required number of hours that they must teach each class, while our employees simply have a minimum total number of required hours they must work.

Cooper and Kingston [9] demonstrated intractability of timetabling problems in assorted ways, the most similar to ours of which is referred to as "intractibility owing to meeting size," which roughly translates to intractability owing to shift length in our problem. But again the parameters and details of the problems they study have meaningful differences from ours. Their timetabling problem is more complicated, requiring multiple sets to be assigned to the "shifts," with the requirement that certain members of set A be placed on the same shift as certain members of set B, in addition to the basic availability constraints. Aloul et al. proposed a SAT-based approach to solve a variant of employee timetabling [1]. In their formulation, employee requirements are defined in terms of minimum days rather than minimum hours, all shifts are considered equal, and they seek to minimize the number of idle workers.

A wide variety of techniques have been used to solve timetabling problems. To name a few, Aloul et al. [1,2] examined Boolean satisfiability and ILP-solvers; Boyer et al. [4] used a branch-and-price algorithm to ensure that employees are only assigned to tasks they are capable of; Elahipanah et al. (2013) use a branch-and-bound search tree [1,2,4,12]. Robinson et al. [22] studied a personnel scheduling problem where a set of tasks must be assigned to a set of employees during specific task intervals with the objective of minimizing labor costs. They, along with a later work [5], propose a network flow solution, but in this setting the employees have already been assigned their shifts, and the flow network is for assigning tasks within that schedule. In our work, we also use a network flow-based algorithm, but our algorithm is used to assign employees to shifts, the problem setting is quite different, and we provide a formal gaurantee on how closely our algorithm approximates an optimal solution.

The scheduling problem most similar to our our own is probably the nurse rostering problem, as it is concerned with fairness to employees [6–8, 17, 19]. Approaches to solving the nurse rostering problem include tabu heuristic search [6]; variable depth search [7]; and heuristic ordering hybridized with a variable neighborhood search [8]. The nurse rostering problem has similar constraints as ESP: there is a set of shifts available for each day (typically "day," "night," and "late night"); an employee has a set availability, and cannot be scheduled when they are unavailable; an employee can only be given one shift per day (this restriction is not necessary for ESP, but we address how to handle this restriction) [19]. The key difference between ESP and the nurse rostering problem is that in ESP, shifts may vary in length, so employee satisfaction is determined based on total hours rather than total number of shifts assigned, and one set of shifts may satisfy an employee while an equal number of shorter shifts may not.

1.2 Contributions

We show that the decision version of ESP is NP-complete. We then present an algorithm that solves a special case of ESP where shifts are of the same length. We further show that for instances that admit a feasible employee schedule (one where $\min_i h_i/r_i \geq 1$), the same algorithm gives approximation guarantees for two variants of our problem: (1) when the "required hours" r_i for each employee i are interpreted to be a minimum number of hours that the employee must work, and additional hours above r_i add to employee satisfaction, and (2) when the "required hours" r_i for each employee i are in fact interpreted to be the *desired hours* of the employee, and hence any additional hours do not add to employee satisfaction, so we cap $\text{OPT} = \min_i h_i/r_i$ at 1.

In the first case, we learn that the further the budget of total available hours k surpasses the total required hours of all employees R, the better the fairness guarantee. And in both cases, keeping all shifts similar in length also improves the fairness guarantee. In particular, in the second case, our algorithm approximates the value of the optimal solution to within an additive

$$\delta \leq \frac{t_{max} - t_{min}}{t_{max}}$$

where t_{min} and t_{max} are the lengths of the shortest and longest shifts, respectively.

However, these results are stipulated by the fact that we allow for a (reasonably bounded amount of) budget overflow. Note that in real world settings t_{min} and t_{max} may not be dramatically different, and the closer they are, the better the approximation guarantee and the lower the budget overflow. Furthermore, in any instance where $k/R \geq t_{max}/t_{min}$, our algorithm is guaranteed to satisfy all employee requirements, i.e., for every employee e_i, $h_i \geq r_i$.

2 Model and preliminaries

The Employee Satisfaction Problem (ESP) can be formalized as follows. We are given as input:

1. A total number of hours available for distribution, k
2. A set of shifts $S = \{s_1, s_2, \dots, s_n\}$, where for each shift $j = 1...n$, we have:
   - a positive integer t_j, the length of shift j

  - a positive integer m_j, the minimum number of employees needed to cover shift s_j
3. A set of employees $E = \{e_1, e_2, \dots, e_m\}$, where for each employee $i = 1...m$, we have:
   - $S_i \subseteq S$, the subset of shifts that employee i is available to work
   - a positive integer r_j, the minimum number of hours that employee e_i must be scheduled to work

We make the following basic assumptions on the input, without which the instance would be trivially infeasible.

1. $k \geq \sum_{s_j \in S} m_j t_j$, i.e., there are at least as many hours in the budget as the shifts require
2. $k \geq \sum_{e_i \in E} r_i$, i.e., there are at least as many hours in the budget as the employees require

For convenience and without loss of generality, we also make the assumption that $r_i \geq t_{min}$ for $i = 1...m$. (If $r_i < t_{min}$ for an employee e_i, we can round r_i up to t_{min} because, since $r_i > 0$, any employee must work at least 1 shift, and it is not possible to assign any employee less than 1 t_{min}-hour shift.)

A *solution* to the problem (or *schedule* or *assignment*) is a mapping $\sigma : E \rightarrow 2^S$ of employees to sets of shifts they are scheduled to work such that the following constraints are met.

1. for any employee e_i, $\sigma(e_i) \subseteq S_i$, [employee availability constraints]
2. for any shift s_j, $|\{e_i : s_j \in \sigma(e_i)\}| \geq m_j$, and [shift requirements]
3. $\sum_{e_i \in E} \sum_{s_j \in \sigma(e_i)} t_j \leq k$ [budget constraint]

We note that we have not yet addressed the issue of overlapping shifts. As currently stated, the problem allows the same employee to be assigned to two shifts that overlap in time. Indeed, the input as specified above does not even include information about which shifts overlap. For the sake of simplifying presentation, we defer our solution to this issue to Section 5.

The total number of hours assigned by schedule σ for employee e_i is denoted

$$h_i = \sum_{s_j \in \sigma(e_i)} t_j$$

(so the third constraint above can be rewritten $\sum_{e_i \in E} h_i \leq k$).

The decision problem ESP-D is to decide whether a schedule exists where all employees work at least their required number of hours, i.e.,

$$\max_{e_i \in E} r_i - h_i \leq 0,$$

or, alternatively,

$$\min_{e_i \in E} h_i / r_i \geq 1.$$

We refer to such schedules as *feasible*.

We highlight both formulations of the objective here because $\max_i r_i - h_i$ may in fact be 0 or negative. Hence, rather than a standard multiplicative approximation to the corresponding optimization problem, we use the second formulation, and give an additive approximation.[1]

Thus, our corresponding optimization objective for ESP will be to assign shifts to employees so as to

[1] A multiplicative approximation of the second objective would be awkward and perhaps misleading since this objective is effectively formulated as a percentage.

$$\text{maximize} \quad \min_{e_i \in E} h_i/r_i.$$

As previously mentioned, we also allow for two interpretations of the input parameters r_i: (1) r_i represents the minimum required hours an employee must work, and satisfaction level h_i/r_i may exceed 1, or (2) r_i represents the maximum number of hours the employee *wishes* to work, and hence satisfaction level h_i/r_i is capped at 1. Formally, the second interpretation yields the following objective, and we refer to this variant of the problem as ESPW.

$$\text{maximize} \quad \min\left\{ \left(\min_{e_i \in E} h_i/r_i \right), 1 \right\}.$$

We provide results for both ESP and ESPW. For the remainder of this paper, we denote $T = \sum_{j=1}^{n} t_j$, $R = \sum_{i=1}^{m} r_i$, $t_{min} = \min_j t_j$, $t_{max} = \max_j t_j$, $r_{min} = \min_i r_i$, and $r_{max} = \max_i r_i$. We use OPT or σ^* to denote the optimal solution, and h_i^* will refer to the total hours assigned to employee i in σ^*. We use $|\sigma|$ to denote the objective function value of the solution σ. We sometimes abuse notation and use an algorithm's name to also refer to the solution it returns. An algorithm A is an *additive δ-approximation* for ESP if $|A| \geq |OPT| - \delta$ for all possible instances of ESP.

2.1 Intractability

We show that the decision version of the ESP problem is NP-complete by reduction from PARTITION.

Theorem 1 ESP-D *is NP-complete*

Proof Given as assignment of employees to shifts it can easily be determined in polynomial time whether $h_i/r_i \geq 1$ for all $e_i \in E$. It remains to show that ESP-D is NP-hard. Recall that the problem PARTITION is defined as follows. Decide whether a set of integers $\{x_1, x_2, \ldots, x_n\}$ can be partitioned into two subsets of equal sum. Given an instance of partition, reduce it to ESP as follows. For each integer x_j in the set of integers, we create a shift of duration t_j, with $m_j = 1$. There are two employees with $r_1 = r_2 = k/2$. Both employees are available to work every shift, and we set $k = T$, the sum of the shift durations. Note that if a feasible schedule of shift assignments exists, then all of the shifts have been assigned exactly once, and so the set of integers has been divided into two subsets of equal sum. If a feasible assignment does not exist, there must be no way to partition the integers. $\square$

3 The algorithm

In this section we look at a special case of ESP-D to provide context and build intuition for our proposed algorithm. Specifically, we demonstrate that if all shifts are the same length, the problem can be solved efficiently. In this case, the problem can be solved in polynomial time by reducing to the circulation problem, which can be reduced to the classic max-flow problem [14].

An instance of the circulation problem is comprised of a flow network $G = (V, \mathcal{E})$, a flow demand value d_v for each node $v \in V$, and a capacity specification $[\ell_e, c_e]$ for each edge $e \in \mathcal{E}$, where ℓ_e is the minimum amount of flow required on edge e and c_e is the maximum

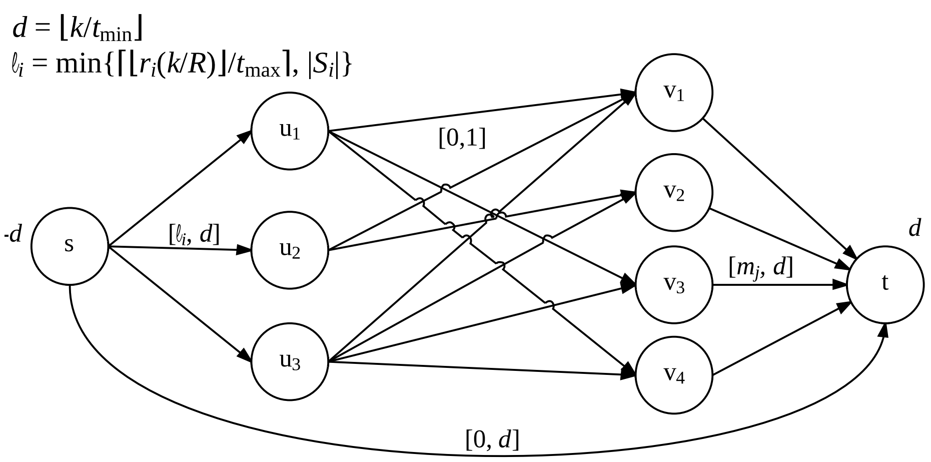
$d = \lfloor k/t_{\min} \rfloor$
$\ell_i = \min\{\lceil \lfloor r_i(k/R) \rfloor / t_{\max} \rceil, |S_i|\}$
-d
s
$[\ell_i, d]$
u1
u2
u3
[0,1]
v1
v2
v3
v4
$[m_j, d]$
t
d
$[0, d]$

Figure 1 An example of the graph G with 3 employees and 4 shifts. Note that the special case of $t_{min} = t_{max}$ is under consideration in this section.

capacity of edge e. Flow must pass along the edges such that the demands specified at each node are satisfied, and capacity constraints on the edges are observed. Demand on the node v is satisfied if $(\text{flow into } v) - (\text{flow out of } v) = d_v$. A *feasible circulation* is a flow where all the edge capacity bounds are observed and the demands on each node are satisfied.

Our network (see Figure 1) has an underlying bipartite graph structure with “employee nodes” on one side and “shift nodes” on the other. A unit of flow from an employee node u_i to a shift node v_j means the employee e_i is assigned to shift s_j.

We now present the algorithm that we will be analyzing for the general case where shifts may be differing lengths, keeping in mind that in this section, we assume all shift lengths are equal, hence $t_{min} = t_{max}$. We let $d = \lfloor k/t_{min} \rfloor$, as this is the maximum number of shifts that can be assigned without exceeding k when $t_{min} = t_{max}$. For each employee e_i, the minimum number of shifts that could satisfy their requirement r_i is $\lceil r_i/t_{max} \rceil$. Therefore, we use this as the basis for the lowerbound on the edge incident to the employee node. Finally, we scale each lowerbound by k/R to ensure that any excess hours will be distributed. Formally, the graph G is constructed as stated in Algorithm 1.

We round down $r_i k/R$ in the lowerbound expression to make sure that we do not overestimate our demand and preclude a feasible solution. Finally, we define the algorithm CIRC-D here as Algorithm 2.

The circulation problem can be efficiently solved by reducing it to the max-flow problem and using, for example, the classic Edmonds-Karp algorithm [11] which has a runtime of $O(|V|^2|\mathcal{E}|)$, or, in the context of our problem, $O((n+m)^2(nm))$. (Better run-times can of course be gained by using any of the series of successive improvements to the run time of solving this classic problem.)

The proof of the following theorem may be found in the full version of this paper.

Theorem 2 *If $t_{max} = t_{min}$, the algorithm* CIRC-D *correctly solves the problem* ESP-D.

```
Algorithm 1: Reducing ESP to the Circulation Problem
  Data: a set of employees $E$, a set of shifts $S$, the shift availability $S_i$ of each employee $i$, and a total
        number of hours $k$
  Result: A circulation flow network G
  Add a "source" node $s$ and a "sink" node $t$;
  for each shift $s_j$ in $S$ do
      Add a node $v_j$ to the "right" side of $G$;
      Add an edge $(v_j, t)$ with capacity bounds $[m_j, d]$;
  end
  for each employee $e_i$ in $E$ do
      Add a node $u_i$ to the "left" side of $G$;
      Add an edge $(s, u_i)$ with capacity bounds $[\min\{\lceil \lfloor r_i k/R \rfloor / t_{max} \rceil, |S_i|\}, d]$ ;
      for each shift $s_j \in S_i$ do
          Add an edge $(u_i, v_j)$ with capacity bounds [0,1];
      end
  end
  Add an edge $(s,t)$ with capacity $[0, d]$;
  Give $s$ a demand value of $-d$ and $t$ a demand value of $d$;
  Give all other nodes a demand value of 0;
```

```
Algorithm 2: Circ-D
  Data: ESP inputs
  Result: Whether or not there is a feasible employee schedule
  1 Follow the procedure in Algorithm 1 to construct the network $G$ using the relevant inputs to the
    ESP-D instance;
  2 Run a circulation solver on the graph $G$;
  3 if there is a feasible circulation then
  4     output YES;
  5 else
  6     output NO;
  7 end
```

4 Additive approximation guarantee

We use Algorithm 2 to approximate an optimal solution to the general ESP (where t_{min} and t_{max} are not necessarily equal), save for the following modifications, and we refer to the resulting algorithm as CIRC:

- In step 2 we return the circulation itself
- In step 3 we construct the assignment of employees to shifts by adding a shift s_j to the set $\sigma(e_i)$ for each edge (u_i, v_j) that has one unit of flow in the circulation. We then return the assignment σ.

The circulation is the same as in Section 3; however, in the general case that $t_{min} < t_{max}$, the lowerbound produced by $\lceil \lfloor r_i k/R \rfloor / t_{max} \rceil$ on each edge (s, u_i), $i = 1...m$, may be fewer than the minimum number of shifts that could satisfy the requirement r_i, and thus no longer guarantees that $h_i \geq r_i$. The budget restrictions are also effectively relaxed with $d = \lfloor k/t_{min} \rfloor$, which is now a potentially loose upperbound on the number of shifts the budget can afford, and can allow more than k hours of shifts to be assigned. These effectively loosened restrictions ensure that, if no feasible circulation is found, then no feasible assignment of employees exists.

The proof of the following lemma may be found in the full version of this paper.

Lemma 1 *If a feasible solution exists for an instance of ESP-D, a feasible circulation can be found (by* CIRC*) in G.*

Of course, the algorithm may return a feasible circulation when there is no feasible assignment of employees to shifts, and it may indeed return an assignment that exceeds the budget of k, which we will also provide worst-case bounds on. But Lemma 1 ensures that if the algorithm does not return a solution, no feasible solution exists, which perhaps indicates to the employer that the input values are unreasonable and must be reconsidered.

We note that the flow demand value $d = \lfloor k/t_{min} \rfloor$ is the minimum possible that still ensures there will be a feasible circulation when there is a feasible assignment. Indeed, if $d < \lfloor k/t_{min} \rfloor$, there are instances with a feasible assignment where the circulation is infeasible.

As an interesting sidenote, we now show that if $\lfloor k/R \rfloor \geq t_{max}/t_{min}$, all employee requirements are guaranteed to be satisfied by CIRC.

Proposition 1 *If $\lfloor k/R \rfloor \geq t_{max}/t_{min}$, then the solution returned by* CIRC *ensures that $h_i \geq r_i$ for all $i = 1...m$.*

Proof Due to the lowerbounds on edges out of s, each employee e_i is guaranteed to be assigned at least $\lceil \lfloor r_i k/R \rfloor / t_{max} \rceil \geq \lceil \lfloor k/R \rfloor r_i / t_{max} \rceil$ shifts. Thus, we have $h_i \geq \lceil \lfloor k/R \rfloor r_i / t_{max} \rceil \cdot t_{min}$. And with $\lfloor k/R \rfloor \geq t_{max}/t_{min}$, then $h_i \geq \lceil r_i t_{min} \rceil / t_{min} \geq r_i$. □

This simple fact may imply a practical rule of thumb for employers: they should have a budget of at least $k >= (t_{max}/t_{min})R$ total hours for distribution if they wish to guarantee that employees can all work the number of hours they are required to.

Before proceeding with proving our guarantee on minimum employee satisfaction, we first show that the budget overflow of CIRC can be reasonably small when (1) shift lengths are all close in size (i.e., $t_{max} - t_{min}$ is small), or (2) there is a large number of t_{min}-length shifts in S.

Let n_{min} be the number of shifts of length t_{min}. The proof of the following lemma may be found in the full version of this work.

Lemma 2 *In the solution returned by* CIRC*, the budget k will not be exceeded by more than* $b = t_{max} \cdot (\lfloor k/t_{min} \rfloor - n_{min}) + t_{min} \cdot n_{min} - k$

$$\approx k(t_{max}/t_{min} - 1) - n_{min}(t_{max} - t_{min}).$$

As a possible rule of thumb for employers: the budget overflow is lower when the number and duration of maximum-length shifts is lower.

We now move onto the approximation guarantee for minimum employee satisfaction. Let $|\text{CIRC}|$ denote the objective function value of our algorithm's solution. We start by giving a lowerbound on the quality of our algorithm's solution (the proof of which may be found in the full version of this work).

Lemma 3 *For any instance of ESP,*

$$|\text{CIRC}| \geq \frac{\lceil r_{max} \lfloor k/R \rfloor / t_{max} \rceil \cdot t_{min}}{r_{max}}$$

Theorem 3 *Assuming there is a feasible solution to* ESP-D*, and allowing for a budget overflow of b, the algorithm* CIRC *provides an additive δ-approximation to* ESP*, where*

$$\delta \leq \frac{T}{r_{max}} - \frac{\lfloor k/R \rfloor t_{min}}{t_{max}}$$

Proof In any instance, the most hours that any employee can have is T: in the case where they are assigned to every existing shift. Therefore, for any employee e_i we have $h_i/r_i \leq T/r_i$; and hence the minimum satisfaction over all employees in OPT is at most T/r_{max}.

Combining this with Lemma 3:

$$\begin{aligned}|\text{OPT}| - |\textsc{Circ}| &\leq \frac{T}{r_{max}} - \frac{\lceil r_{max} \lfloor k/R \rfloor / t_{max} \rceil \cdot t_{min}}{r_{max}} \\ &\leq \frac{T}{r_{max}} - \frac{\lfloor k/R \rfloor \cdot t_{min}}{t_{max}}\end{aligned}$$

□

We defer to the full version of this work the proof of the following theorem, which shows that the above guarantee on the performance of Circ for ESP (Theorem 3) is essentially tight.

Theorem 4 *There is an instance of ESP where*

$$|\text{OPT}| - |\textsc{Circ}| = \delta \geq \frac{T}{r_{max}} - \frac{\lceil r_{max} \lfloor k/R \rfloor / t_{max} \rceil t_{min}}{r_{max}}$$

In the case of the problem ESPw (where satisfaction levels h_i/r_i are always capped at 1, which would be the case when the r_i inputs represent employees' maximum desired hours rather than minimum required hours), we immediately arrive at the following simple characterization of the guarantee of Circ.

Theorem 5 *Assuming there is a feasible solution to ESP-D, and allowing for a budget overflow of b, the algorithm* Circ *provides an additive δ-approximation to* ESPw, *where*

$$|\text{OPT}| - |\textsc{Circ}| = \delta \leq \frac{t_{max} - t_{min}}{t_{max}}$$

Proof By Lemma 3, and since $\lfloor k/R \rfloor \geq 1$, we have

$$|\textsc{Circ}| \geq \frac{\lfloor k/R \rfloor t_{min}}{t_{max}} \geq t_{min}/t_{max}.$$

By definition of ESPw we know that $|OPT| \leq 1$, hence $|\text{OPT}| - |\textsc{Circ}| \leq 1 - t_{min}/t_{max}$.
□

We now demonstrate that Theorem 5 is tight.

Theorem 6 *There is an instance of* ESPw *where*

$$\delta \geq \frac{t_{max} - t_{min}}{t_{max}}$$

Proof The lowerbound on our algorithm for ESPw is demonstrated by a worst-case instance described as follows (also see Figure 2). The instance has $m = 4$ employees, $n = m + 1 = 5$ shifts, and $k = T = 29$. Employee requirements and availability are: $r_1 = 7, S_1 = \{s_1, s_2, s_3\}$; $r_2 = 5, S_2 = \{s_2, s_3\}$; $r_3 = 5, S_3 = \{s_1, s_4\}$; $r_4 = 7, S_4 = \{s_4, s_5\}$.

Shift lengths and are alternating: $t_1 = 5$, $t_2 = 7$, $t_3 = 5$, $t_4 = 7$, and $t_5 = 5$, and shift requirements are $m_j = 1$ for $j = 1 \dots 5$. The lowerbound for each employee edge (s, u_i), $i = 1 \dots m$, is hence 1. $d = \lfloor k/t_{min} \rfloor = 5$, which means there are 5 shifts available for distribution among the employees. In the optimal solution, the shifts are assigned as illustrated in Figure 2 for an objective function value of $|OPT| = 1$. However, another feasible circulation exists (as illustrated) that does not satisfy all employees' required hours, giving a minimum satisfaction of t_{min}/t_{max}. Hence $|OPT| - |\textsc{Circ}| = \delta \geq 1 - t_{min}/t_{max}$. □

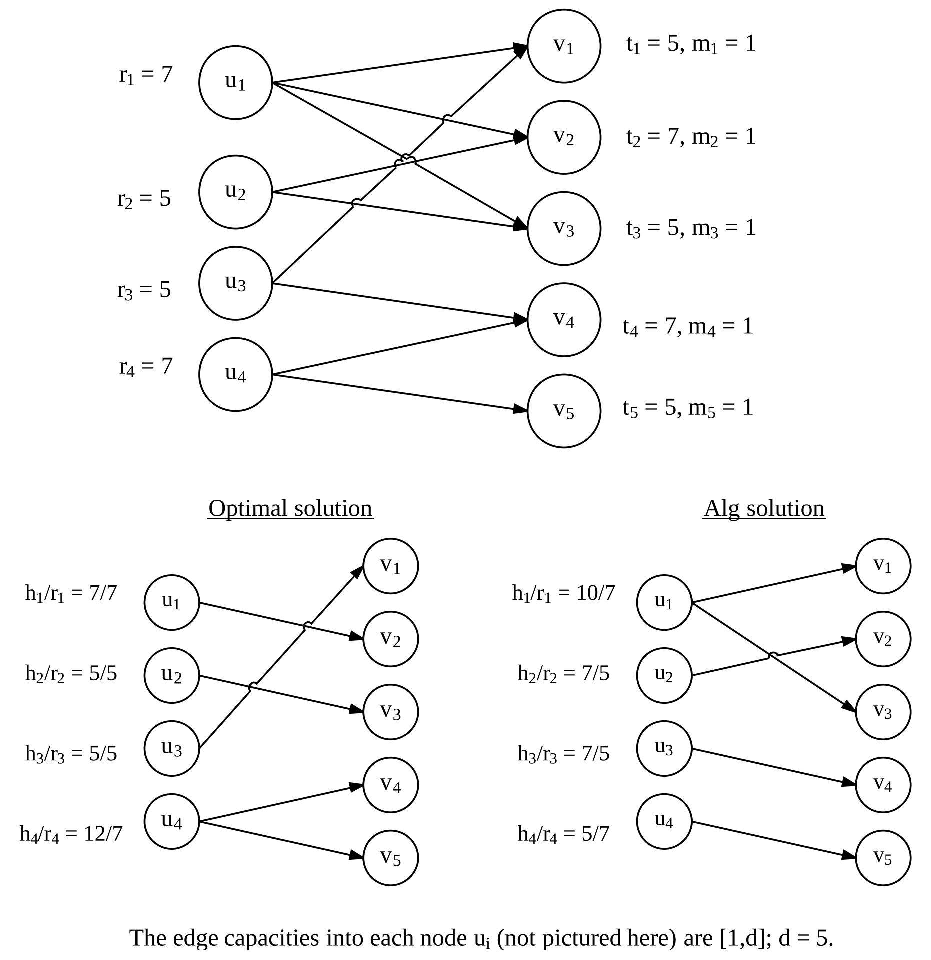

Figure 2 An instance that demonstrates the lowerbound of δ for ESPw.

5 Overlapping shifts

Our algorithm as presented thus far assumes that shifts do not overlap. Any two shifts assigned to an employee must not share any hours or else the assignment is invalidated. In order to resolve this and ensure that overlapping shifts are not assigned to the same employee, the following change can be applied to the circulation design.

For each employee e_i, for every pair of overlapping shifts in $\{s_x, s_y\} \in S_i$:

1. Remove the two edges (u_i, v_x) and (u_i, v_y).
2. Add a "median" node w_i "between" the corresponding shift nodes v_x and v_y as follows.
   Add an edge from u_i to the median node w_i.
   Add edges from the median node w_i to each of the two shift nodes v_x and v_y.
   Set all edges to and from w_i to have capacity [0,1].
3. If either of the shift nodes v_x or v_y now has two or more of these median nodes adjacent to it (emanating from u_i), further modify the graph as follows. For each such shift node v_j, $j \in \{x, y\}$, with adjacent median nodes $(w_{i_1} \dots w_{i_\mu})$:

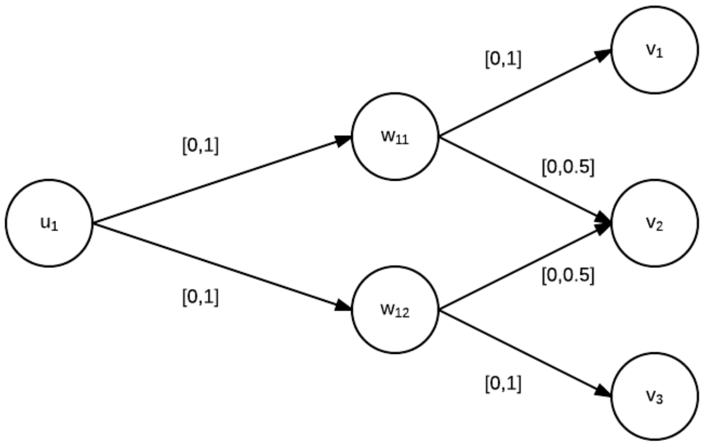

Figure 3 Shifts s_1 and s_2 overlap, and shifts s_2 and s_3 overlap, but shifts s_1 and s_3 are not in conflict

```
For each edge $(w_{i_k}, v_j)$, $k = 1 \dots \mu$:
  Set its capacity to $[0, 1/\mu]$
```

An example output of this adjusted procedure is illustrated in Figure 3. In this example, shifts s_1 and s_2 overlap, and shifts s_2 and s_3 overlap, but shifts s_1 and s_3 are not in conflict.

The algorithm must further be modified to prefer whole flows to fractional ones in its tie-breaking; an available edge with capacity of 1 should be preferred over an available edge with a fractional capacity. For example, in Figure 3, there are many different flows that will saturate both edges leaving u_1, but the one that sends whole units of flow over the edges (w_{11}, v_1) and (w_{12}, v_3) is preferred.

The assignments of employees to shifts is determined as before: an employee i is assigned to a shift j if and only if there is one unit of flow from node u_i to node v_j. In particular, if there is less than 1 unit of flow from an employee node to a shift node, that employee is not assigned to that shift.

With this additional procedure, the guarantees of the algorithm remain the same, but now it is certain that no employee will be scheduled for overlapping shifts.

6 Conclusion

Our work shifts the focus of employee timetabling problems onto employee satisfaction. We present an approximation algorithm for the egalitarian objective of maximizing minimum employee satisfaction. ESP can be applied to many types of work environments where varying weekly schedules are used. It addresses the concerns of both the management and the employees: while we allow some budget overflow, we provide a bound for the overflow amount, which the employer can make use of in setting their initial budget (k) value; employees are guaranteed a lowerbound on how many hours they will be given relative to what they need, which promises that the schedule will be relatively fair; and all shift requirements are satisfied, ensuring that every shift will have adequate coverage.

The quality of the guarantees are dependent on the input, specifically on the size difference between the shortest and longest shift in the set, the maximum employee requirement, and the size of k.

Some important future directions include: (1) finding an algorithm with a better approximation guarantee, (2) considering the more complex problem of allowing both a minimum and maximum amount hours to be specified for each employee, and (3) considering the employee's r_i values to be private information that must be extracted from the employee truthfully (making it a mechanism design problem).

References

1. Aloul, F., Al-Rawi, B., Al-Farra, A., & Al-Roh, B. "Solving the employee timetabling problem using boolean satisfiability." in *2006 Innovations in Information Technology*, November 19-21, 2006, Dubai, 4085403 (2006).
2. Aloul, F., Zahidi, S., Al-Farra, A., & Al-Roh, B. "Solving the employee timetabling problem using advanced SAT & ILP techniques." *Journal of Computers*, Vol. 8(4), pp. 851-858, 2013.
3. Artigues, C., Gendreau, M., Rousseau, L.M., & Vergnaud, A. "Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound." *Computers and Operations Research*, Vol. 36(8), pp. 2330-2340, 2009.
4. Boyer, V., Gendron, V., & Rousseau, L. "A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem." *Journal of Scheduling*, pp. 1-13, 2013.
5. Brucker, P., & Qu, R. "Network flow models for intraday personnel scheduling problems." *Annals of Operations Research*, pp. 1-8, 2012.
6. Burke, E., Cowling, P., De Causmaecker, P., Vanden Berghe, G. "A Memetic Approach to the Nurse Rostering Problem." *Applied Intelligence*, Vol. 15(3), pp. 199-214, 2001.
7. Burke, E., Curtois, T., Qu, R., & Vanden Berghe, G. "A Time Pre-defined Variable Depth Search for Nurse Rostering." Technical Report, University of Nottingham, 2007.
8. Burke, E., Curtois, T., De Causmaecker, P., Post, G., Qu, R., Vanden Berghe, G. & Veltman, B. "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem." *European Journal of Operational Research*, Vol. 188(2), pp. 330-341, 2008.
9. Cooper, T. B., & Kingston, J. H. (1996, October). The Complexity of Timetable Construction Problems. In *Practice and Theory of Automated Timetabling*, Edinburgh, UK, August 1995.
10. Dowsland, K. "Nurse scheduling with tabu search and strategic oscillation." *European Journal of Operational Research*, Vol. 106, pp. 393-407, 1998.
11. Edmonds, J., & Karp, M. "Theoretical improvements in algorithmic efficiency for network flow problems." *Journal of the Association for Computing Machinery*, pp. 248-264, 1972.
12. Elahipanah, M., Dulniers, G., & Lacasse-Guay, E. "A two-phase mathematical- programming heuristic for flexible assignment of activities and tasks to work shifts." *Journal of Scheduling*, pp. 1-18, 2013.
13. Even, S., Itai, A., & Shamir, A. "On the complexity of timetable and multicommodity flow problems." *SIAM J. Comput.*, Vol. 5(4), pp. 691-703, 1976.
14. Ford, L. R., and Delbert Ray Fulkerson. Flows in networks. Vol. 1962. Princeton University Press: Princeton, 1962.
15. Gotlieb, C. C. (1963, January). The construction of class-teacher time-tables. In *IFIP congress* (Vol. 62, pp. 73-77).
16. Gotlieb, C. C. (1962, January). The construction of class-teacher time-tables. In COMMUNICATIONS OF THE ACM (Vol. 5, No. 6, pp. 312-313).
17. Holmes, H., Pierskalla, W., & Rath, G. "Nurse Scheduling Using Mathematical Programming." *Operations Research*, Vol. 24(5), 1976.
18. Kragelund, L. "Solving a timetabling problem using hybrid genetic algorithms." *Software - Practice and Experience*, Vol. 27, pp. 1121-1134, 1997
19. Maenhout, B. & Vanhoucke, M. "Comparison and hybridization of crossover operators for the nurse scheduling problem." *Annals of Operations Research*, Vol 159, pp.333-353, 2007.
20. Meisels, A., Gudes, E., & Soloterevsky, G. "Combining rules and constraints for employee timetabling." *Int'l Journal of Intelligent Systems*, Vol.12, pp.419-439, 1997.
21. Meisels, A., & Shaerf, A. "Modelling and solving employee timetabling problems." *Annals of Mathematics and Artificial Intelligence*, Vol. 39(1-2), pp. 41-59, 2003.
22. Robinson, R., Sorli, R., Zinder, Y. (2005). Personnel scheduling with time windows and preemptive tasks. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, Pittsburgh, August 2004.