A PTAS for Unsplittable Flow on a Path

Fabrizio Grandoni*
fabrizio@idsia.ch
IDSIA, USI-SUPSI
Lugano, Switzerland

Tobias Mömke†
moemke@informatik.uni-augsburg.de
University of Augsburg
Augsburg, Germany

Andreas Wiese‡
a.wiese@vu.nl
Vrije Universiteit
Amsterdam, Netherlands

ABSTRACT

In the Unsplittable Flow on a Path problem (UFP) we are given a path with edge capacities, and a set of tasks where each task is characterized by a subpath, a demand, and a weight. The goal is to select a subset of tasks of maximum total weight such that the total demand of the selected tasks using each edge e is at most the capacity of e. The problem admits a QPTAS [Bansal, Chakrabarti, Epstein, Schieber, STOC'06; Batra, Garg, Kumar, Mömke, Wiese, SODA'15]. After a long sequence of improvements [Bansal, Friggstad, Khandekar, Salavatipour, SODA'09; Bonsma, Schulz, Wiese, FOCS'11; Anagnostopoulos, Grandoni, Leonardi, Wiese, SODA'14; Grandoni, Mömke, Wiese, Zhou, STOC'18], the best known polynomial time approximation algorithm for UFP has an approximation ratio of $1 + \frac{1}{e+1} + \epsilon < 1.269$ [Grandoni, Mömke, Wiese, SODA'22]. It has been an open question whether this problem admits a PTAS. In this paper, we solve this open question and present a polynomial time $(1 + \epsilon)$-approximation algorithm for UFP.

CCS CONCEPTS

• **Theory of computation** → **Dynamic programming**; **Packing and covering problems**; **Rounding techniques**.

KEYWORDS

approximation, unsplittable flow, dynamic programming

ACM Reference Format:
Fabrizio Grandoni, Tobias Mömke, and Andreas Wiese. 2022. A PTAS for Unsplittable Flow on a Path. In *Proceedings of the 54rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3519959

1 INTRODUCTION

In the Unsplittable Flow on a Path problem (UFP), we are given an undirected path $G = (V, E)$ with a capacity $u(e) \in \mathbb{N}$ for each edge $e \in E$ and a set of n tasks T. Each task $i \in T$ is specified by a subpath $P(i) \subseteq E$, a demand $d(i) \in \mathbb{N}$, and a weight (or profit) $w(i) \in \mathbb{N}$. For each $e \in E$ we denote by T_e be the set of all tasks $i \in T$ such that $P(i)$ contains e. For $T' \subseteq T$, let $w(T') := \sum_{i \in T'} w(i)$ and $d(T') := \sum_{i \in T''} d(i)$. The goal is to select a subset $OPT \subseteq T$ of tasks of maximum total weight $opt = w(OPT)$ such that for each edge e the total demand of the selected tasks using e does not exceed $u(e)$, i.e., $d(OPT \cap T_e) \leq u(e)$.

UFP is a very well-studied problem, see e.g., [6]. It has many applications and connections to settings like multi-commodity demand flow [19], caching [21], independent set of rectanges [12], bandwidth allocation [8, 20, 33], and resource allocation [7, 13, 23, 38]. For example, the reader might interpret the path as a time interval subdivided into time slots (the edges). We have a given amount of a resource that varies over time (the edge capacities). Each task can be executed in a specific time interval, and demands for a fraction of this resource.

Since UFP is a generalization of Knapsack, it is weakly NP-hard, and in fact it is even strongly NP-hard [12, 21]. Therefore, approximation algorithms have been studied for this problem. The first non-trivial (polynomial-time) approximation algorithm for UFP by Bansal, Friggstad, Khandekar, and Salavatipour has an approximation factor of $O(\log n)$ [6]. The first constant approximation factor was achieved by Bonsma, Schulz, and Wiese [12]. This constant was later improved to $2 + \epsilon$ by Anagnostopoulos, Grandoni, Leonardi, and Wiese [4], to $5/3 + \epsilon$ by Grandoni, Mömke, Wiese, and Zhou [30], and very recently to $1 + 1/(e + 1) + \epsilon < 1.269$ by Grandoni, Mömke, and Wiese [28]. It is an important open question whether UFP admits a PTAS. This seems likely since there is a QPTAS for it [5, 11, 27] and there are PTASs for several special cases [11, 29].

The mentioned 1.269-approximation algorithm [28] uses the approach to phrase the problem as a solitary game, on which we build in this paper. We describe here a simplified version of this game, and we assume first that all input edges have a uniform capacity of some value U, and that we are allowed to use a slightly larger capacity of $(1 + \epsilon)U$ (resource augmentation). Note that even for this simplified setting, there is no better approximation ratio known than 1.269 (and achieving a PTAS for this case is an open problem).

There is a global linear order of the input tasks in which they are sorted from left to right by their respective start edges (i.e., the respective leftmost edges of each path $P(i)$). The tasks are presented to a player in this order. Given a task i, the player can decide to select or to reject i. If she selects i, then she gains a profit of $w(i)$. After that, in case that $d(i) < \delta U$, for some small constant $\delta > 0$, (we say that i is *small* in this case), we round the demand of i randomly to either 0 or δU, such that in expectation the rounded demand equals the true demand $d(i)$. Otherwise, we say that i is *large* and we do not change the demand of i. Then, the next task is presented to the player. While playing the game, the player needs to

*Partially supported by the SNSF Excellence Grant 200020B_182865/1.
†Partially supported by DFG Grant 439522729 (Heisenberg-Grant) and DFG Grant 439637648 (Sachbeihilfe)
‡Partially supported by ANID Fondecyt Regular grant 1200173.

STOC '22, June 20–24, 2022, Rome, Italy

ACM ISBN 978-1-4503-9264-8/22/06…$15.00
https://doi.org/10.1145/3519935.3519959

ensure that the selected tasks use at most $(1+\epsilon)U$ units of capacity on each edge, according to their rounded demands. Her goal is to play the game in order to maximize her profit (in expectation).

Due to the ordering of the tasks, we can compute the optimal strategy for this game in polynomial time by a simple dynamic program (DP). Also, it is relatively easy to show that there exists a strategy whose expected profit is at least $(1-\epsilon)opt$. One such strategy is to discard all the tasks not in OPT, and to select each task $i \in OPT$ unless it does not fit together with the previously selected tasks (according to their rounded demands). One can show that the latter happens only with probability at most ϵ if δ is small enough.

Therefore, we would get a PTAS (with resource augmentation) if we could transform the computed optimal strategy into a solution to the given UFP instance, by losing only a factor of $1+\epsilon$. However, this is not straightforward. For example, it is not sufficient to simply simulate the player playing the game and selecting all tasks that she selects. It might happen that the player selects a task i that does not fit together with the previously selected tasks on some edge $e \in P(i)$ according to the *original* demands, however, they fit according to the *rounded* demands since many small tasks using e had been rounded down (more than expected).

Grandoni, Mömke and Wiese [28] use a routine to transform the optimal player's strategy into a feasible solution that loses a constant factor of the profit obtained by the player from small tasks (while retaining essentially all the profit from large ones). In this paper, we define a more sophisticated game and an associated transformation procedure, that allow us to preserve almost all the profit from all tasks.

1.1 Our Contribution

We present a PTAS for (the general case of) UFP. For simplicity let us start with the uniform capacity case with resource augmentation. Our algorithm is again based on a game, however, it is not a game with a globally fixed order in which the tasks are presented to the player. Instead, if the player selects a task i that is large or a rounded up small task, we split the remaining game into two subgames that are then played independently by the player. In particular, this partitions the remaining tasks (that have not yet been presented to the player) into two sets, one for each of the subgames. The two subgames and in particular this partitioning are defined based on $P(i)$. If the player selects a small task i and i is rounded down, we do *not* split the remaining game into two subgames. Therefore, even if we fix a strategy and play the game twice according to this strategy, the sequence of these partitions can be very different, since the small tasks are rounded randomly. One way of visualizing the different subgames (that arise when the player plays the game) is a rooted tree in which each internal vertex has two children.

The game is designed such that the total number of possible subgames is bounded by a polynomial in n. Thus, again we can compute the optimal strategy to play the game in polynomial time with a DP. Also, we can again show that there is a strategy for the player that achieves at least a profit of $(1-\epsilon)opt$ in expectation, by selecting only the tasks in OPT and rejecting a task $i \in OPT$ if too many previously selected small tasks intersecting $P(i)$ were rounded up, i.e., many more than expected (which happens with probability ϵ).

The hard part is to transform a given (optimal) strategy for the player into a feasible solution to the given UFP instance. Like for the game in [28], it is not sufficient to simply simulate the player and select every task that she selects: it might be that she selects a set of tasks that exceeds the capacity on some edge e, while she is allowed to select them since many more tasks than expected are rounded down. Also, we cannot fix this problem by simply rejecting a task if it does not fit, since it might contribute a lot to the expected profit of the player.

Instead, we still simulate the player, but we reject tasks selected by her in a non-trivial way: we reject a task i if *after* selecting i the player selects many other small tasks that use two carefully chosen edges of $P(i)$, and among them too many are rounded down. Also, we reject a task i if *before* selecting i on each edge of $P(i)$ too many previously selected small tasks were rounded down.

We need to bound the loss due to the rejected tasks. First, we argue that for a selected task i it is very unlikely that i is rejected due to the first case above, and hence in expectation the resulting loss is negligible. For the second case, we construct a fractional solution x in which for each task i the value x_i is the probability that task i is selected by the player but rejected by us (due to the second case). The key step is to argue that x uses only an $O(\epsilon)$ fraction of the capacity of each edge (for all tasks together). Therefore, we can compute a solution with small tasks only, placed in the additional space due to the resource augmentation, which compensates for this loss.

The hardest part is to show that the remaining tasks obey the edge capacities. To do this, we crucially use some properties of our game that the game in [28] does not have. In particular, our game is carefully designed such that this part of the argumentation works and thus we can transform the optimal strategy into a feasible solution with a small loss.

In order to obtain a PTAS without resource augmentation (and for arbitrary edge capacities) we employ the slack lemma from [11, 29] which implies that there are $(1+\epsilon)$-approximate solutions in which a part of the edge capacities remains unused. This slack capacity can then be used in a similar way as the extra capacity due to resource augmentation, analogously to previous work on UFP [28, 30].

1.2 Further related work

UFP admits an FPT-$(1+\epsilon)$ approximation scheme for the unweighted case where the parameter is the cardinality of the optimum solution [39] (see also [34]). Our result subsumes [39] up to polynomial factors in the running time. The natural LP-formulation for UFP suffers from an integrality gap of $\Omega(n)$ [15], hence some research was devoted to finding stronger LP relaxations [3, 18].

The Bag UFP problem is the generalization of UFP where tasks are partitioned into subsets (bags), and at most one task per bag can be selected [14, 26]. Another generalization of UFP, the Unplittable Flow on a Tree problem (UFT), is defined such that instead of a path we are given a tree in the input [18, 24] (see also [1] for a further generalization of the latter result). The Storage Allocation problem is a variant of UFP with additional geometric packing constraints among tasks [9, 10, 36, 37].

UFP turns out to be connected in a non-obvious way (see [12]) to the Maximum Independent Set of Rectangles problem [2, 16, 17, 22, 25, 35] and to the Highway problem [31, 32].

1.3 Organization of this paper

The rest of this paper is organized as follows. In Section 2 we give some preliminary lemmas and definitions. We will describe in some detail a PTAS with resource augmentation for the case of uniform capacities in Sections 3-5. This will allow us to illustrate most of our novel ideas (besides solving an open problem on its own). In more detail, in Section 3 we will describe a game for this case, and give an overview of how it is used to derive the PTAS. In Section 4 we describe a *strategy* of the player with expected profit close to the optimum profit. Finally in Section 5 we show how to *round* a given strategy in order to obtain a solution with large enough expected profit. The proofs of Sections 2-5 which are omitted due to space constraints will be provided in a journal version of this paper (however an expert reader might be able to reconstruct most of them). In Section 6 we will sketch how to obtain a PTAS (without resource augmentation) for general edge capacities. Here unfortunately we have to omit most of the technical details, part of which are non-trivial: the missing details will be provided in a journal version.

2 PRELIMINARIES

Unless differently stated, we next assume that we are given an instance of UFP with uniform edge capacity, i.e., $u(e) = U$ for each edge $e \in E$ for some value $U \in \mathbb{N}$. Our goal is to describe a PTAS under resource augmentation, i.e., for any given $\epsilon > 0$ we compute a solution with profit at least $(1 - O(\epsilon))opt$ that uses capacity at most $(1 + O(\epsilon))U$ on each edge.

We imagine the edges of the input path graph G as sorted from left to right. For two distinct edges e and f, let $e < f$ and $f > e$ denote that e appears to the left of f. Analogously, $e \leq f$ (and $f \geq e$) if $e < f$ or $e = f$. By standard reductions we can assume that on each edge there is at most one task starting or one task ending, which will simplify algorithms and proofs. Similarly we can assume that each edge is the leftmost or rightmost edge of some task, and hence $|E| \leq 2n$.

Given a sub-path P' of G, we let $\ell(P')$ and $r(P')$ be the leftmost and rightmost edge of P', resp. We say that P' starts at $\ell(P')$ and ends at $r(P')$. We sometimes simply denote by P' the edges $E(P')$ of a path P'. We say that P' spans P'' if $P'' \subseteq P'$. For a task i, we let $\ell(i) = \ell(P(i))$ and $r(i) = r(P(i))$. Similarly, we use the path terminology in the framework of tasks meaning implicitly the corresponds paths. For instance, we might say that task i spans task j if $P(j) \subseteq P(i)$ and that task i ends to the right of task j if $r(i) > r(j)$ etc.

Let $\epsilon > 0$ such that $1/\epsilon \in \mathbb{N}$. We define $\delta := \epsilon^5$. Thanks to resource augmentation, we can assume that the input instance has some additional properties by standard reductions (see e.g. [28] and references therein).

Lemma 1. *Suppose there is a polynomial-time ρ-approximation algorithm with $1 + O(\epsilon)$ resource augmentation for an instance of UFP with uniform capacity U where the following holds:*

- *for each task $i \in T$ we have that $d(i) = 1$ or $d(i)$ is an integral multiple of $\delta^2 U \in \mathbb{N}$ with $d(i) \geq \delta U$,*
- *$U = O(\frac{n}{\epsilon})$ where n is the number of tasks.*

Then there is a polynomial-time $\rho(1+O(\epsilon))$-approximation algorithm with $1 + O(\epsilon)$ resource augmentation for an instance of UFP with uniform capacity U.

We assume the properties due to Lemma 1 from now on. Notice that $\delta U \in \mathbb{N}$ since $\delta^2 U \in \mathbb{N}$ and $\frac{1}{\delta} = \frac{1}{\epsilon^5} \in \mathbb{N}$. Similarly, $\epsilon U \in \mathbb{N}$. We call a task i *large* if $d(i) \geq \delta U$ and *small* otherwise. Let T^{large} and T^{small} be the set of large and small tasks, respectively.

3 THE GAME

In order to construct our PTAS for uniform edge capacities and resource augmentation, we will define a solitary game. In this game the player is presented tasks in a given (adaptive) order. Each time a task i is presented, she decides whether to select or reject i (plus some additional technical decisions defined later). After the selection of a task i, the *rounded demand* of i on each edge e is revealed to the player. This is a random quantity defined according to certain rules. The player has to guarantee (deterministically) that the total rounded demand of the selected tasks on each edge e is at most $\tilde{U} = (1 + O(\epsilon))U$, and her goal is to maximize the expected profit of the selected tasks. A *strategy* of the player is a way to play this game. We will prove the following facts:

(1) An optimal strategy of the player, i.e., one achieving the maximum possible expected profit play*, can be computed in polynomial time;
(2) There exists a strategy with expected profit play $\geq (1 - O(\epsilon))opt$;
(3) Given a strategy with expected profit play, in polynomial time one can compute a UFP solution with expected profit at least $(1 - O(\epsilon))$play and using (actual) capacity at most $\tilde{U}+O(\epsilon)U$ on each edge (we call this a *rounding of the player's strategy*).

Combing the above facts, a PTAS for UFP with uniform capacities and resource augmentation easily follows.

The very first step of the game is to compute an optimal solution T_{LP} of small tasks only, assuming that each edge $e \in E$ has a capacity of only ϵU. We do this using the following lemma which in fact solves a standard LP for UFP (which is integral for unit-demand tasks). The subscript LP of T_{LP} refers to this.

Lemma 2. *In polynomial time we can compute an optimal solution to an instance of UFP defined by input tasks T^{small}, the path E, and edge capacities of $\tilde{u}(e) = \epsilon U$ for each $e \in E$.*

Proof. We can formulate this problem by the following linear program

$$\begin{aligned} \max \quad & \sum_{i \in T^{small}} w(i) x_i \\ \text{s.t.} \quad & \sum_{i \in T^{small} \cap T_e} d(i) \cdot x_i \leq \epsilon U \;\; \forall e \in E \\ & 0 \leq x_i \leq 1 \quad \forall i \in T^{small}. \end{aligned}$$

Its constraint matrix is totally unimodular since $d(i) = 1$ for each small task, by the consecutive ones property, and other basic properties of totally unimodular matrices. Also, note that $\epsilon U \in \mathbb{N}$ since $\delta U \in \mathbb{N}$. Therefore, we can compute an optimal solution to the LP in polynomial time, which yields an optimal solution to our problem. □

After that, the game is played recursively in *subgames*. We define $U' := (1+\epsilon)U$. The input for each subgame consists of

(1) a subpath $E' \subseteq E$, partitioned into a left subpath E'_L and a right subpath E'_R;
(2) a set of input tasks T', all fully contained in E', such that each $i \in T'$ uses at least one edge of E'_R;
(3) edge capacities $u' : E' \to \{0, 1, \ldots, U'\}$ satisfying the following properties:
   (a) restricted to E'_L, u' is non-decreasing and has at most $\Gamma := \frac{1+\epsilon}{\delta^2}$ steps;
   (b) restricted to E'_R, u' is non-increasing, has at most Γ steps, and its entries are integral multiples of $\delta^2 U$.

The root subgame is $(T', E'_L, E'_R, u') = (T \setminus T_{LP}, \emptyset, E, u_{U'})$ where $u_{U'}(e) = U'$ for each edge $e \in E$. We assume that there is a global variable gain indicating the current profit of the player, which initially has the value gain $= 0$.

We now describe how a subgame is played. Suppose that we are given a subgame (T', E'_L, E'_R, u'). If $T' = \emptyset$ then the subgame stops immediately. Otherwise, we define a task $i \in T'$ as follows.

- if there is a task $i' \in T'$ such that $P(i')$ uses some edge of E'_L, namely $P(i') \cap E'_L \neq \emptyset$, then we define i to be the task in T' with *rightmost end edge* with this property;
- otherwise, we define i to be the task in T' with *leftmost start edge*.

We do not allow the player to select task i if the following happens:

- i is large and there exists an edge $e \in P(i)$ with $u'(e) < d(i)$ or
- i is small and there exists an edge $e \in P(i) \cap E'_L$ with $u'(e) = 0 < d(i) = 1$ or an edge $e \in P(i) \cap E'_R$ with $u'(e) < \delta U$.

In this case we say we *skip* i and we continue recursively with the subgame $(T' \setminus \{i\}, E'_L, E'_R, u')$. Otherwise, we say that i is *presented* to the player and she can choose to select i or to reject it. If she rejects i, then we also continue recursively with the game $(T' \setminus \{i\}, E'_L, E'_R, u')$ (as if we had skipped i).

Rounding task demand. Now suppose that the player selects i. Then she gains a profit of $w(i)$ (i.e., the variable gain is increased by that amount). Then we define the *rounded demand* $d_{round}(i, e)$ of i on each edge e as follows. If i is small, we flip a biased coin and define i to be *rounded up* with probability $p_{up} := \frac{1}{\delta U}$ and *rounded down* otherwise. Then

$$d_{round}(i,e) := \begin{cases} 0 & \text{if } e \notin P(i); \\ d(i) & \text{if } i \text{ is large and } e \in P(i); \\ 1 & \text{if } i \text{ is small and } e \in E'_L \cap P(i); \\ \delta U & \text{if } i \text{ is small and rounded up,} \\ & \text{and } e \in E'_R \cap P(i); \\ 0 & \text{if } i \text{ is small and rounded down,} \\ & \text{and } e \in E'_R \cap P(i). \end{cases}$$

Notice that $d_{round}(i, \cdot)$ is the demand of i on each edge if i is large. Otherwise, this is true only for edges in $P(i) \cap E'_L$ or not in $P(i)$, while on the remaining edges $e \in P(i) \cap E'_R$, $d_{round}(i,e)$ is equal to $d(i) = 1$ only in expectation. For a subset of tasks T', $d_{round}(T', e) := \sum_{i \in T'} d_{round}(i, e)$. We obtain a new capacity function u'' from u' by removing the rounded demand of i.

$$u''(e) := u'(e) - d_{round}(i,e) \quad \forall e \in E'.$$

Observe that u'' might not be non-decreasing on E'_L. To fix this, we decrease the capacity $u''(e)$ on each edge $e \in E'_L$ if there is some edge $f \in E'_L$ on the right of e with smaller capacity. We do not modify u'' along E'_R. Formally (see also Figure 1), we define

$$u''_{\text{fix}}(e) := \begin{cases} \min_{f \in E'_L : f \geq e} u''(f) & \text{if } e \in E'_L; \\ u''(e) & \text{if } e \in E'_R. \end{cases}$$

Note that this operation does not restrict us in our future selections of tasks, since if some task j with $P(j) \cap E'_L \neq \emptyset$ is presented later, then $P(j)$ uses the rightmost edge of E'_L. Therefore, if for some $e \in P(j) \cap E'_L$ we have that $u''(e) > u''_{\text{fix}}(e) = \min_{f \in E'_L : e \geq f} u''(f)$, then the edge f achieving the latter minimum is the bottleneck for j on E'_L.

The reader might have noticed that u''_{fix} might not be non-increasing on E'_R if i is large or rounded up. However, it will turn out that this is not problematic due to the way we will define the next subgames.

Recursing in subgames. If i is rounded down (hence small), we continue with the subgame $(T' \setminus \{i\}, E'_L, E'_R, u''_{\text{fix}})$. Assume now that i is rounded up or large (see also Figure 1). We want to partition $T' \setminus \{i\}$ into two sets of tasks T'_L, T'_R and define a subgame for each of them. Let $E'_{RL} = P(i) \cap E'_R$ and $E'_{RR} = E'_R \setminus P(i)$. We define T'_R as the tasks in $T' \setminus \{i\}$ that intersect both E'_{RL} and E'_{RR} (we call such tasks *boundary*) or which are fully contained in E'_{RR}. We call *inner* a task which is not boundary. Let T'_L be the remaining tasks in $T' \setminus \{i\}$. Both sets T'_L and T'_R might contain tasks using edges in E'_{RL}, but all other edges of $E' := E'_L \cup E'_R$ can be used by only one of these sets. In order to create two independent subgames for T'_L and T'_R, the player needs to define a non-decreasing step function $\tilde{u} : E'_{RL} \to \mathbb{N}_0$ that intuitively denotes the amount of capacity on E'_{RL} that is assigned to the tasks in T'_R. We require for each edge $e \in E'_{RL}$ that $\tilde{u}(e)$ is an integral multiple of $\delta^2 U$ with $\tilde{u}(e) \leq u''_{\text{fix}}(e) = u''(e)$.

For our two subgames for T'_L and T'_R we define capacity functions u'_L and u'_R as follows:

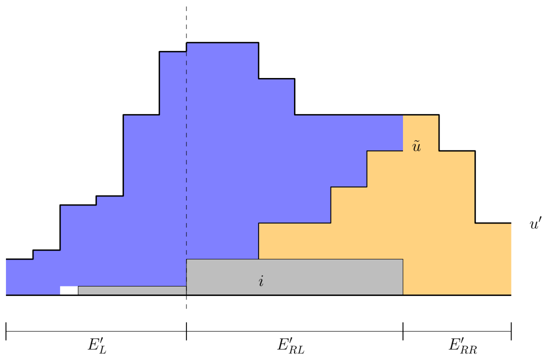

Figure 1: Recursing in subgames. Task i is small and rounded up, resulting in a left subgame $(T'_L, E'_L, E'_{RL}, u'_L)$ and a right subgame $(T'_R, E'_{RL}, E'_{RR}, u'_R)$. The blue and orange areas depict u'_L and u'_R, respectively. The white area corresponds to a region where $u''_{fix}(e) < u''(e)$.

$$u'_L(e) = \begin{cases} u''_{\text{fix}}(e) & \text{if } e \in E'_L; \\ u''_{\text{fix}}(e) - \tilde{u}(e) & \text{if } e \in E'_{RL}. \end{cases} \quad u'_R(e) = \begin{cases} \tilde{u}(e) & \text{if } e \in E'_{RL}; \\ u''_{\text{fix}}(e) & \text{if } e \in E'_{RR}. \end{cases}$$

Notice that $u'_R(e) + u'_L(e) = u''_{\text{fix}}(e)$ for each $e \in E'$ (defining $u'_L(e)$ and $u'_R(e)$ to be zero on edges where they are undefined). We define our *left subgame* by $(T'_L, E'_L, E'_{RL}, u'_L)$ and our *right subgame* by $(T'_R, E'_{RL}, E'_{RR}, u'_R)$.

Lemma 3. *Given that (T', E'_L, E'_R, u') is a valid subgame, the corresponding tuples (i) $(T'\setminus\{i\}, E'_L, E'_R, u')$ (if i is skipped or not selected), (ii) $(T'\setminus\{i\}, E'_L, E'_R, u''_{\text{fix}})$ (if i is rounded down), (iii) $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ and (iv) $(T'_L, E'_L, E'_{RL}, u'_L)$ (if i is large or rounded up) are valid subgames.*

Proof. In all cases property (1) follows immediately.

(i) This case holds since only the tasks have changed.

(ii) It is sufficient to argue that u''_{fix} satisfies (3). For (3.b), observe that, for each $e \in E'_R$, $u''_{fix}(e) = u''(e) = u'(e)$, hence the claim. For (3.a), we notice that by construction u''_{fix} is non-decreasing over E'_L and contains the same number of steps as u' in the same subpath, hence the claim.

(iii) By the order in which tasks are considered, the tasks intersecting both E'_{RL} and E'_{RR} are fully contained in $E'_{RL} \cup E'_{RR}$. The remaining tasks assigned to T'_R are fully contained in E'_{RR}. Hence T'_R satisfies property (2). Let us show that u'_R satisfies property (3). For $e \in E'_{RR}$, $u''_R(e) = u''_{fix}(e) = u'(e)$, hence (3.b) holds. Along E'_{RL}, $u'_R = \tilde{u}$. Hence the entries of u'_R are multiples of $\delta^2 U$ and u'_R is non-decreasing. Furthermore trivially $\tilde{u}(e) \leq u'(e) \leq U'$, hence $\tilde{u}$ can have at most $\frac{U'}{\delta^2 U} = \frac{1+\epsilon}{\delta^2} = \Gamma$ steps.

(iv) By the order in which tasks are considered, the tasks intersecting E'_L are fully contained in $E'_L \cup E'_{RL}$. The remaining tasks assigned to T'_L are fully contained in E'_{RL}. Hence T'_L satisfies property (2). Let us show that u'_L satisfies property (3). For $e \in E'_L$, $u'_L(e) = u''_{fix}(e)$. Property (3.a) follows since, as mentioned in case (ii), u''_{fix} is non-decreasing over E'_L and contains the same number of steps as u'. For $e \in E'_{RL}$, $u'_L(e) = u''_{fix}(e) - \tilde{u}(e) = u''(e) - \tilde{u}(e) = u'(e) - d_{round}(i,e) - \tilde{u}(e)$, where $d_{round}(i,e) = d(i)$ if i is large and $d_{round}(i,e) = \delta U$ if i is small and rounded up. Hence $u'_L(e)$ is the sum of three values which are multiples of $\delta^2 U$. Since over E'_{RL}, $u'(e)$ is non-increasing, $d_{round}(i,e)$ is constant, and $\tilde{u}(e)$ is non-decreasing, u'_L is non-increasing in the same subpath. As a consequence, since its entries are multiples of $\delta^2 U$, u'_L can have at most Γ steps similarly to case (iii). ◻

Then the player plays the left and the right subgame. The two subgames are independent; however, in our analysis it is convenient to assume that the player plays the left subgame before the right one. This imposes a total order on the time tasks are selected by the player: we use the notation $i \prec j$ and $j \succ i$ to indicate that i appears earlier than j in this order. We also use $i \preceq j$ and $j \succeq i$ to denote $i \prec j$ or $i = j$. When the player finishes playing the last subgame (recursively), the current value of the global variable gain defines her total profit from playing the game.

3.1 Strategies and transforming them into solutions

We say that a *strategy* is a function that describes the action of the player in each subgame (T', E'_L, E'_R, u'), i.e., when a task i is presented, whether i is selected or not, and the function $\tilde{u}$ in case that a selected task i is rounded up or large. Given a strategy, let play $:= w(T_{LP}) + \mathbb{E}[gain]$ denote the expected profit of the player (notice that we include also the profit of T_{LP}).

We can compute the *optimal* strategy, i.e. the one with maximum expected profit play*, in polynomial-time by a dynamic program. First of all, we observe that there are at most $n^{O_\epsilon(1)}$ subgames that can be reached by the player after starting with the root subgame. Indeed, it is obvious by definition that there are only polynomially many choices for the parameters E'_L, E'_R and u'. This holds also for T', restricted to the reachable subgames, by the order in which tasks are presented to the player. It is then sufficient to describe the behavior of the player in each reachable subgame. In each such subgame, the player has only a polynomial number of choices. It is then sufficient to select an (optimal) choice which leads to the largest expected profit in terms of the tasks selected in the current subgame and in its descendants. Such an optimal choice can be easily computed via a dynamic program. We omit the (relatively straightforward) details here.

Lemma 4. *In time $n^{O_\epsilon(1)}$ we can compute the optimal strategy and its expected profit* play*.

In Section 4 we will show that there exists a strategy with expected profit close to *opt*.

Lemma 5. *There exists a strategy for the player with expected profit* play $\geq (1 - O(\epsilon))opt$

In Section 5 we show how to transform a given a strategy (e.g., the strategy computed in Lemma 4) into an integral solution with almost the same profit using some extra space, in more detail:

Lemma 6. *Given a strategy of the player with expected profit* play, *there is a randomized algorithm with a running time of $n^{O_\epsilon(1)}$ that* *computes an integral solution $\bar{T}$ such that $\mathbb{E}[w(\bar{T})] \geq (1 - O(\epsilon))$play and $d(\bar{T} \cap T_e) \leq (1 + O(\epsilon))U'$ for each edge $e \in E$.*

We can conclude that:

Theorem 7. *There is a (randomized) PTAS for UFP with uniform capacities U and resource augmentation.*

Proof. It is sufficient to compute the optimal strategy as in Lemma 4, and then convert it into a solution $\bar{T}$ via Lemma 6. By Lemmas 5 and 6 one has $\mathbb{E}[w(\bar{T})] \geq (1-O(\epsilon))\text{play}^* \geq (1-O(\epsilon))(1-O(\epsilon))opt$. The maximum demand of $\bar{T}$ on every edge is at most $(1 + O(\epsilon))U' = (1 + O(\epsilon))(1 + \epsilon)U$. □

4 A PROFITABLE STRATEGY FOR THE PLAYER

In this section we prove Lemma 5, i.e., we show that there exists a strategy for the player with expected profit play $\geq (1 - O(\epsilon))opt$.

In our strategy, we never select a task $i \in T \setminus OPT$. Suppose that a task $i \in OPT$ is presented to the player in some subgame (T', E'_L, E'_R, u') (so in particular i is not skipped). Intuitively, in this subgame and in the subgames into which we recurse from it, we want to select the tasks $T' \cap OPT$ using the edge capacities given by u'. We select i if $u'(r(i)) \geq d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$ and otherwise we reject i. Intuitively, when we reject i then among the previously selected tasks from OPT using $r(i)$ much more were rounded up than expected and hence they need much more space than in OPT. We reject i in order to compensate for this, i.e., i needs space in OPT but not in our game. However, we will show later that we reject a task i with probability at most $O(\epsilon)$ due to our resource augmentation.

If i is small and rounded down then we simply continue with the next subgame. Suppose now that i is rounded up or large. Recall that E'_{RR} denotes the edges of E'_R on the right of $P(i)$ and $E'_{RL} = P(i) \cap E'_R$. Also, recall that T'_R are the tasks in $T' \setminus \{i\}$ that intersect both E'_{RL} and E'_{RR} or that are fully contained in E'_{RR}, i.e., that end in E'_{RR}. We define OPT_R to be the tasks in $OPT \cap T'$ that (like T'_R) intersect both E'_{RL} and E'_{RR} or that are fully contained in E'_{RR}. We need to define the non-decreasing step function $\tilde{u} : E'_{RL} \to \mathbb{N}_0$ which intuitively represents the capacity assigned on E'_{RL} to tasks in T'_R. For each edge $e \in E'_{RL}$ we define $\tilde{u}(e) := \min\left\{\left\lceil \frac{|OPT_R \cap T_e|}{\delta^2 U} \right\rceil \delta^2 U, u''_{fix}(e)\right\}$. Taking the minimum here guarantees us that we do not assign too much capacity to $\tilde{u}$. Recall that we recurse on the subgames $(T'_L, E'_L, E'_{RL}, u'_L)$ and $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ where in particular u'_L and u'_R are defined based on $\tilde{u}$.

Let play denote the expected profit of this strategy.

4.1 Analysis of our strategy

We need to prove two properties of our strategy. First, we need to show that it constructs well-defined subgames. Second, we need to show that play $\geq (1 - \epsilon)opt$.

We start by showing the first property. If in a subgame (T', E'_L, E'_R, u') no large task is selected and no task is (selected and) rounded up, then clearly the player recurses again in a well-defined subgame. Consider now the other subgames.

Lemma 8. *Let (T', E'_L, E'_R, u') be a subgame that arises when the player plays according to our strategy in which a large task is selected or a small task is (selected and) rounded up. Then the resulting tuples $(T'_L, E'_L, E'_{RL}, u'_L)$ and $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ are valid subgames.*

Proof. Consider a subgame (T', E'_L, E'_R, u') in which (according to our strategy) the player selects a small task i which is rounded up. All tasks in OPT_R end in E'_{RR}. Thus, the function $f : E'_{RL} \to \mathbb{N}_0$ defined by $f(e) = \left\lceil \frac{|OPT_R \cap T_e|}{\delta^2 U} \right\rceil \delta^2 U$ is a non-decreasing step-function whose entries are all integral multiples of $\delta^2 U$. Also, the function $g : E'_{RL} \to \mathbb{N}_0$ defined by $g(e) = u''(e) = u'(e) - \delta U$ has only steps that are integral multiples of $\delta^2 U$. Thus, u''_{fix} is a non-decreasing step-function whose entries are all integral multiples of $\delta^2 U$. Thus, since $\tilde{u}(e) = \min\{f(e), u''_{\text{fix}}(e)\}$ for each $e \in E'_{RL}$, we have that $\tilde{u}$ is a non-decreasing step function on E'_{RL} whose entries are all integral multiples of $\delta^2 U$. Since $|OPT_R \cap T_e| \leq U$, hence $f(e) \leq U$, $\tilde{u}$ has at most $1/\delta^2$ steps.

The case where the player selects a large task i is analogous, with δU replaced by $d(i)$ (which is still an integral multiple of $\delta^2 U$). □

We address now the second point mentioned above by proving the following lemma.

Lemma 9. *Each task $i \in OPT$ is selected by the player with probability at least $1 - O(\epsilon)$.*

Lemma 9 implies directly that play $\geq (1 - O(\epsilon))opt$. In the remainder of this section we prove Lemma 9. Consider a task $i \in OPT$ for which we want to prove Lemma 9. First, we show that i is presented to the player or skipped exactly once (so in particular there will always be a subgame in which i is the next task to be considered).

Lemma 10. *There is exactly one subgame in which $i \in OPT$ is presented to the player or skipped.*

Proof. We prove by induction that, given a subgame (T', E'_L, E'_R, u'), if $i \in T'$ then i is presented to the player or skipped exactly once in (T', E'_L, E'_R, u') or in a game at which we arrive recursively from (T', E'_L, E'_R, u'). For the base case, the claim is clearly true if $T' = \emptyset$.

Suppose that we start a subgame (T', E'_L, E'_R, u') such that $i \in T'$. Suppose that a task i' is presented to the player or skipped. Assume that afterwards we recurse on one subgame $(T' \setminus \{i'\}, E'_L, E'_R, u')$. If $i \neq i'$ the claim follows by induction since then $i \in T' \Leftrightarrow i \in T' \setminus \{i'\}$. If $i = i'$ then $i \in T'$ and i was presented to the player or skipped in the subgame (T', E'_L, E'_R, u'). This implies $i \notin T' \setminus \{i'\}$: the claim follows by induction from $(T' \setminus \{i'\}, E'_L, E'_R, u')$. Assume now that after presenting i' to the player we recurse on two subgames $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ and $(T'_L, E'_L, E'_{RL}, u'_L)$ (in this case i' was not skipped). Similarly as before, if $i \neq i'$ the claim follows by induction since then $i \in T' \Leftrightarrow i \in T'_L \cup T'_R$. Otherwise (i.e., if $i = i'$), i was presented to the player or skipped in the subgame (T', E'_L, E'_R, u'). This implies $i \notin T'_L \cup T'_R$: the claim follows since in the two subgames and in their descendent subgames only tasks in $T'_L \cup T'_R$ might be presented or skipped. □

Assume now that i is skipped or presented to the player but not selected. Due to the monotonicity of the profile u', this is due to the leftmost edge $\ell(i)$ of $P(i)$ or the rightmost edge $r(i)$ of $P(i)$ since these are the edges with smallest capacity u' in $E' \cap P(i)$.

Lemma 11. *If $i \in OPT$ is skipped or presented to the player but not selected, we have that $u'(\ell(i)) < d(i)$ or $u'(r(i)) < d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$.*

Proof. By the definition of the game and our strategy, task i is skipped or rejected if one of the following conditions holds:

(1) i is small and there is an edge $e \in P(i) \cap E'_L$ with $u'(e) = 0$,
(2) i is small and there is an edge $e \in P(i) \cap E'_R$ with $u'(e) < \delta U$,
(3) i is large and there is an edge $e \in P(i)$ with $u'(e) < d(i)$,
(4) $u'(r(i)) < d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$.

If condition 1 holds then also $u'(\ell(i)) < d(i)$ holds since u' is non-decreasing on E'_L.

If condition 2 holds then $u'(r(i)) < \delta U$ since u' is non-increasing on E'_R. The claim follows since $\delta U \le \epsilon U/3 \le d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$.

If condition 3 holds and the corresponding edge e is in E'_L then $u'(\ell(i)) < d(i)$ since u' is non-decreasing on E'_L. If condition 3 holds and the corresponding edge e is in E'_R then $u'(r(i)) < d(i)$ since u' is non-increasing on E'_R. The claim follows since $d(i) \le d(OPT \cap T' \cap T_{r(i)}) < d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$, where the first inequality holds since $i \in OPT \cap T' \cap T_{r(i)}$. □

We argue now that only with probability ϵ one of the conditions of Lemma 11 applies in the subgame in which i is considered, i.e., in which i is selected, skipped, or rejected. In fact, for the first condition $u'(e) < d(i)$ for some edge $e \in E'_L$ we prove even that it never happens. Intuitively, the reason is that on the edges in E'_L each selected task i has deterministically demand $d(i)$, and when we defined the capacities of the edges in E'_L we ensured that they are sufficient for the corresponding tasks from OPT.

Lemma 12. *In the subgame (T', E'_L, E'_R, u') in which i is considered, it cannot happen that there is an edge $e \in P(i) \cap E'_L$ such that $u'(e) < d(i)$.*

Proof. Assume by contradiction that there is an edge $e \in P(i) \cap E'_L$ such that $u'(e) < d(i)$. Observe that in this case i must be a boundary task since otherwise $P(i) \cap E'_L = \emptyset$. Let $(\bar{T}, \bar{E}_L, \bar{E}_R, \bar{u})$ be the last played subgame for which $i \in \bar{T}$ and $e \in \bar{E}_R$ (such a subgame must exist since the root subproblem satisfies this property). Thus, there must be a task $j \in \bar{T}$ that was selected in that subproblem and that was large or small and rounded up. Also, it must be that $r(j)$ lies on the right of $\ell(i)$ by the definition of $(\bar{T}, \bar{E}_L, \bar{E}_R, \bar{u})$.

Since j was selected, we have that $\bar{u}(r(j)) \ge d(OPT \cap \bar{T} \cap T_{r(j)}) + \epsilon U/3$. Let $\bar{E}_{RL} := P(j) \cap \bar{E}_R$ and $\bar{E}_{RR} := \bar{E}_R \setminus \bar{E}_{RL}$. Let $OPT_R \subseteq OPT \cap \bar{T}$ denote the tasks in $OPT \cap \bar{T}$ that start in $\bar{E}_{RL}$ and end in $\bar{E}_{RR}$. In particular, for the profile $\tilde{u} : \bar{E}_{RL} \to \mathbb{N}_0$ that the player defined after i' was rounded up, it holds that $\tilde{u}(r(i')) = \left\lceil \frac{|OPT_R \cap T_{r(j)}|}{\delta^2 U} \right\rceil \delta^2 U$ (so the minimum of the definition of $\tilde{u}(r(j))$ is attained in that value). Note that the profile $\bar{u}$ is non-increasing on $\bar{E}_{RL}$. Also, each task in OPT_R using $\ell(i)$ also uses $r(j)$. Therefore, for each edge $\bar{e} \in \bar{E}_{R,L}$ on the right of $\ell(i)$, the profile $\tilde{u}$ yields enough capacity to accomodate all tasks in $OPT_R \cap T_{\bar{e}}$, i.e., we have that $\tilde{u}(\bar{e}) = \left\lceil \frac{|OPT_R \cap T_{\bar{e}}|}{\delta^2 U} \right\rceil \delta^2 U$. Therefore, in each subgame $(\hat{T}, \hat{E}_L, \hat{E}_R, \hat{u})$ played after $(\bar{T}, \bar{E}_L, \bar{E}_R, \bar{u})$ for which $i \in \hat{T}$ holds, on each edge $\hat{e} \in P(i) \cap \hat{E}_L$ we have that $d(OPT \cap \hat{T} \cap T_{\hat{e}}) \le \hat{u}(\hat{e})$ since on the edges $\hat{E}_L$ the task demands are *not* rounded randomly. This holds in particular for (T', E'_L, E'_R, u'), thus contradicting the assumption that there is an edge $e \in P(i) \cap E'_L$ such that $u'(e) < d(i)$. □

Finally, we show that the second condition of Lemma 11 happens only with probability ϵ. We will show that when this happens, then during the game the small tasks in $T_{r(i)} \cap OPT$ selected before i are rounded such that their total (rounded) demand is much higher than expected. To quantify this, for a task j and an edge e we define $disc(j, e)$, the *discrepancy* of j on e, to be the difference of the actual demand of j on e and its rounded demand $d_{round}(i, e)$ on e. Hence, this difference is zero if j is large, j does not use e, or $e \in E'_L$ in the subgame in which j was selected. Formally, assuming that j was selected in some subgame (T', E'_L, E'_R, u'), we define

$$disc(j, e) := \begin{cases} 0 & \text{if } e \in P(j) \cap E'_L, e \notin P(j), \text{or } j \text{ is large;} \\ 1 - \delta U & \text{if } e \in P(j) \cap E'_R \text{ and } j \text{ is rounded up;} \\ 1 & \text{if } e \in P(j) \cap E'_R \text{ and } j \text{ is rounded down;} \end{cases}$$

For a set of tasks T' we define $disc(T', e) := \sum_{j \in T'} disc(j, e)$.

Intuitively, in expectation the discrepancy of a set of tasks T' is zero on each edge e. If it is much higher (resp., lower) than zero, then many more tasks using e than expected were rounded down (resp., rounded up). We say that $disc(T', e)$ is *large (positive)* if it is at least $\frac{\epsilon}{2}U$ and *large negative* if it is at most $-\frac{\epsilon}{2}U$.

Definition 13 (Prefixes). *Fix a strategy of the player and let $e \in E$. Suppose that the player plays the game and let $i_1, ..., i_k$ be the small tasks from T_e that are selected by her, in this order. We say that e has a prefix with large (resp., large negative) discrepancy if, for some $k' \le k$, $disc(\{i_1, \ldots, i_{k'}\}, e)$ is large (resp., large negative).*

Hence, e has a prefix with large negative (resp., positive) discrepancy if in this prefix the number of rounded up small tasks is by more than $\frac{\epsilon}{2}U \cdot \frac{1}{\delta U} = \frac{1}{2\epsilon^4}$ larger (resp., smaller) than expected. However, we show in the next lemma that this is unlikely.

Lemma 14. *Fix a strategy of the player and let $e \in E$. The edge e has a prefix with large or large negative discrepancy with probability at most ϵ^2.*

Proof. Let S be the sequence of tasks selected by the player in T_e, in order of selection. We define a different game which is equivalent to the original one. Let B be a sequence of random bits where each bit is set to one with probability $p_{up} = \frac{1}{\delta U}$. During the game, when the k-th task is added to S, round it up iff the k-th bit of B has value one. Notice that tasks in S are rounded with the correct probability.

Let $Q = \frac{1+\epsilon}{\delta}$ be an upper bound on the number of small tasks using a given edge which can be rounded up. For each $j \in \{1, \ldots, k\}$ with $k := \frac{8(1+\epsilon)}{\epsilon}$, let B_j be the prefix of B of length $j\frac{\epsilon U}{4}$ and $o(B_j)$ be the number of ones in B_j. Denote by $\mathcal{B}_j$ the event that $o(B_j) \le j\frac{\epsilon U}{4} \cdot p_{up} - \frac{\epsilon^2}{4\delta}$ (and hence B_j has discrepancy larger than $\frac{\epsilon^2}{4\delta} \cdot \delta U = \frac{\epsilon^2 U}{4}$) or that $o(B_j) \ge j\frac{\epsilon U}{4} \cdot p_{up} + \frac{\epsilon^2}{4\delta}$ (and hence B_j has discrepancy smaller than $-\frac{\epsilon^2 U}{4}$). Let also $\mathcal{B}$ denote the event that $o(B_k) < Q$.

We claim that if none of the events $\mathcal{B}_1, \ldots, \mathcal{B}_k, \mathcal{B}$ happens, then each prefix of S has neither large nor large negative discrepancy. Assume that $\mathcal{B}$ does not happen. Then the player will never need to use bits beyond B_k since within the Q-th one bit the capacity of the considered edge e is necessarily saturated by rounded up tasks. This means that S has a prefix of large positive or negative discrepancy if and only if B_k has. However, for each $j \in \{1, \dots, k\}$ we have that since neither $\mathcal{B}_{j-1}$ nor $\mathcal{B}_j$ happens, all the prefixes of length between $(j-1)\frac{\epsilon U}{4}$ and $j\frac{\epsilon U}{4} - 1$ have discrepancy at most $\frac{\epsilon^2 U}{4} + \frac{\epsilon^2 U}{4} - 1 < \epsilon^2 U$ and at least $-\frac{\epsilon^2 U}{4} - \frac{\epsilon^2 U}{4} + 1 > -\epsilon^2 U$. Thus, if none of the events $\mathcal{B}_1, \dots, \mathcal{B}_{k-1}, \mathcal{B}$ happens, then S has no prefix of large positive or negative discrepancy.

It is therefore sufficient to show that $Pr[\mathcal{B} \cup \mathcal{B}_1 \cup \dots \cup \mathcal{B}_k] \leq Pr[\mathcal{B}] + \sum_{j=1}^{k} Pr[\mathcal{B}_j] \leq \epsilon^2$. The expected number of ones in $\mathcal{B}_j$ is $\mu_j := j\frac{\epsilon U}{4} \cdot p_{up} = j\frac{\epsilon}{4\delta}$. Hence by Chernoff's bound

$$
\begin{aligned}
&Pr[\mathcal{B}_j] \\
&= Pr[o(B_j) \leq j\frac{\epsilon U}{4} p_{up} - \frac{\epsilon^2}{4\delta}] + Pr[o(B_j) \geq j\frac{\epsilon U}{4} p_{up} + \frac{\epsilon^2}{4\delta}] \\
&= Pr[o(B_j) \leq j\frac{\epsilon}{4\delta} - \frac{\epsilon^2}{4\delta}] + Pr[o(B_j) \geq j\frac{\epsilon U}{4} p_{up} + \frac{\epsilon^2}{4\delta}] \\
&= Pr[o(B_j) \leq \mu_j(1 - \frac{\epsilon}{j})] + Pr[o(B_j) \geq \mu_j(1 + \frac{\epsilon}{j})] \\
&\leq 2e^{-\frac{\epsilon^2}{3j^2}\mu_j} = 2e^{-\frac{\epsilon^3}{12j\delta}} \leq 2e^{-\frac{\epsilon^3}{12k\delta}} = 2e^{-\frac{\epsilon^4}{12\delta \cdot 8(1+\epsilon)}} \leq \epsilon^4
\end{aligned}
$$

if ϵ is sufficiently small since $\delta \leq \epsilon^5$. Also, since

$$\mathbb{E}[o(B_k)] = k\frac{\epsilon}{4\delta} = \frac{2(1+\epsilon)}{\delta} = 2Q,$$

for ϵ small enough

$$
\begin{aligned}
Pr[\mathcal{B}] &= Pr[o(B_k) < Q] \leq Pr[o(B_k) < \mathbb{E}[o(B_k)] \cdot (1 - \frac{1}{2})] \\
&\leq e^{-\frac{1}{8}2Q = e^{-\frac{1+\epsilon}{4\delta}}} \leq \epsilon^4.
\end{aligned}
$$

Therefore, by the union bound, the probability that one of the events $\mathcal{B}_1, \dots, \mathcal{B}_k, \mathcal{B}$ happens is at most $k \cdot \epsilon^4 + \epsilon^4 \leq \epsilon^2$. □

Finally, we show that if some task $i \in OPT$ is rejected because of the second condition of Lemma 11, then $r(i)$ has a prefix with large negative discrepancy, which by Lemma 14 is unlikely. If a task j is selected in a subgame $(\hat{T}, \hat{E}_L, \hat{E}_R, \hat{u})$, then we define $P_L(j) := P(j) \cap \hat{E}_L$ and $P_R(j) := P(j) \cap \hat{E}_R$.

Lemma 15. *Consider a subgame (T', E'_L, E'_R, u') that is played according to an arbitrary strategy, let $\hat{T}_S$ the small tasks selected before the player plays this subgame, and let $UP(\hat{T}_S) \subseteq \hat{T}_S$ denote the tasks in $\hat{T}_S$ that were rounded up. Let i be the task that is presented to the player in (T', E'_L, E'_R, u'). Then*

- *for each $j \in \hat{T}_S$ and each $e \in P(j) \cap E'_R$ we have that $e \in P_R(j)$,*
- *for each task $j \in UP(\hat{T}_S)$ we have that $E'_R \subseteq P(j)$ or $E'_R \cap P(j) = \emptyset$,*
- *for each $j \in \hat{T}_S$ for which $P(j)$ uses an edge $e \in E'_R$ such that $e \in P(i)$ or e lies on the right of $P(i)$, we have that $P(j)$ uses the leftmost edge of $P(i) \cap E'_R$.*

Proof. We prove this lemma by induction on the subgames. The claim is clearly true for the initial subgame $(T \setminus T_{LP}, \emptyset, E, u_{U'})$ since then $\hat{T}_S = \text{UP}(\hat{T}_S) = \emptyset$.

Suppose by induction that the claim is true for some subgame (T', E'_L, E'_R, u') in which a task $i \in T'$ is considered, and let $\hat{T}_S$ denote the small tasks selected before the player starts this game.

If i is skipped or rejected by the player or it is selected and rounded down, then the first two claims of the lemma hold immediately for the next subgame $(T' \setminus \{i\}, E'_L, E'_R, u')$. We want to prove the third claim. Let i' be the task that is presented to the player in $(T' \setminus \{i\}, E'_L, E'_R, u')$. Consider a task $j \in \hat{T}_S$ for which $P(j)$ uses an edge $e \in E'_R$ such that $e \in P(i')$ or e lies on the right of $P(i')$. Since i was presented to the player before i', we have that $e \in P(i)$ or e lies on the right of $P(i)$. By the induction hypothesis, $P(j)$ uses the leftmost edge of $P(i) \cap E'_R$. This edge is identical to or lies on the left of the leftmost edge of $P(i') \cap E'_R$, again since i was presented to the player before i'. Thus, $P(j)$ uses the leftmost edge of $P(i') \cap E'_R$. In case that i was selected and rounded down, then the task i appears in the set of small tasks selected before the player starts the game $(T' \setminus \{i\}, E'_L, E'_R, u')$ (i.e., the "new" set $\hat{T}_S$). And indeed, if i uses an edge e for which $e \in P(i')$ or e lies on the right of $P(i')$, then, since i was presented to the player before i', we conclude that $P(i)$ uses the leftmost edge of $P(i') \cap E'_R$.

Suppose now that i is selected and i is large or small and rounded up. Then the first two claims hold for the subgame $(T'_L, E'_L, E'_{RL}, u'_L)$ since $E'_{RL} \subseteq E'_R$ and $E'_{RL} \subseteq P(i)$ and $(T'_L, E'_L, E'_{RL}, u'_L)$ is played before $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ which is the other subgame that we recurse into after (T', E'_L, E'_R, u'). We want to prove the third claim now for the subgame $(T'_L, E'_L, E'_{RL}, u'_L)$. Let i' be the task that is presented to the player in $(T'_L, E'_L, E'_{RL}, u'_L)$. For each task $j \in \hat{T}_S$ the claim follows by induction: if $P(j)$ uses an edge $e \in E'_{RL}$ such that $e \in P(i')$ or e lies on the right of $P(i')$, then also $e \in P(i)$ or e lies on the right of $P(i)$, and like above $P(j)$ uses the leftmost edge of $P(i') \cap E'_{RL}$. For $j = i$ it follows since $P(i)$ uses each edge $e \in E'_{RL}$ that is used by $P(i')$ or lies on the right of $P(i')$ since i was presented to the player before i'.

Also, the first two claims hold for the subgame $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ since $E'_{RR} \subseteq E'_R$ and $E'_{RR} \cap P(i) = \emptyset$. We want to prove the third claim for the subgame $(T'_R, E'_{RL}, E'_{RR}, u'_R)$. For each task $j \in \hat{T}_S$ the claim follows by induction: let i'' be the task that is presented to the player in $(T'_R, E'_{RL}, E'_{RR}, u'_R)$. If $P(j)$ uses an edge $e \in E'_R$ such that $e \in P(i'')$ or e lies on the right of $P(i'')$, then also $e \in P(i)$ or e lies on the right of $P(i)$, and like above $P(j)$ uses the leftmost edge of $P(i') \cap E'_{RR}$. Also, we observe that i does not use any edge of E'_{RR}, so the case $j = i$ does not appear here. □

Lemma 16. *If i is considered in a subgame (T', E'_L, E'_R, u') such that $u'(r(i)) < d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3$ then $r(i)$ has a prefix with large negative discrepancy.*

Proof. Assume that i is considered in a subgame (T', E'_L, E'_R, u') with

$$u'(r(i)) < d(OPT \cap T' \cap T_{r(i)}) + \epsilon U/3. \quad (1)$$

Since OPT is a feasible solution, we have that $d(OPT \cap T_{r(i)}) \leq U$. Let $\widetilde{OPT} \subseteq OPT \cap T_{r(i)}$ denote the tasks in $OPT \cap T_{r(i)}$ that were selected by the player before starting the subgame (T', E'_L, E'_R, u'). Consider all previously played subgames in which the player selected a task $i^{(j)} \in T_{r(i)}$ that is large or small and rounded up. The edge $r(i)$ lies on the right part of each such task $i^{(j)}$: if $i^{(j)}$ is small and rounded up, this follows from Lemma 15, and if $i^{(j)}$ is large this can be shown similarly as in Lemma 15. Let $OPT_R^{(1)}, \dots, OPT_R^{(k)}$ be their respective sets OPT_R and let $\tilde{u}_1, \dots, \tilde{u}_k$ be their respective functions $\tilde{u}$. Recall that by construction for each j we have that

$$\tilde{u}_j(r(i)) \leq d(OPT_R^{(j)} \cap T_{r(i)}) + \delta^2 U. \tag{2}$$

Since each considered $i^{(j)}$ has (rounded) demand at least δU on $r(i)$,

$$k \leq \frac{U'}{\delta U} \leq \frac{1+\epsilon}{\delta}. \tag{3}$$

Now observe that in OPT the edge $r(i)$ is used by the following sets of tasks:

- previously selected (small) tasks in $\widetilde{OPT} \cap T^{small}$,
- previously selected (large) tasks in $\widetilde{OPT} \cap T^{large}$,
- tasks in $OPT \cap T' \cap T_{r(i)}$ that are not (yet) selected when (T', E'_L, E'_R, u') is played,
- tasks in $\bigcup_{j=1}^k OPT_R^{(j)} \cap T_{r(i)}$ that are not (yet) selected when (T', E'_L, E'_R, u') is played.

Since all tasks in OPT use a capacity of at most U on $r(i)$, we have that

$$d\left(\widetilde{OPT} \cap T^{small}\right) + d\left(\widetilde{OPT} \cap T^{large}\right) + \sum_{j=1}^{k} d(OPT_R^{(j)} \cap T_{r(i)}) + d\left(OPT \cap T' \cap T_{r(i)}\right) \leq U.$$

Since $d(i) = 1$ for each small task, this implies that

$$|\widetilde{OPT} \cap T^{small}| \leq U - d(OPT \cap T' \cap T_{r(i)}) - d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^{k} d(OPT_R^{(j)} \cap T_{r(i)}). \tag{4}$$

By Lemma 15, for each task $i' \in \widetilde{OPT} \cap T^{small}$ we have that $r(i) \in P_R(i')$. Thus, if after selecting a task $i' \in \widetilde{OPT} \cap T^{small}$ in the game the available capacity $u'(r(i))$ decreased, this was because i' was rounded up. Thus, the total capacity of $r(i)$ used by rounded up tasks in $i' \in \widetilde{OPT} \cap T^{small}$ is $U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^k \tilde{u}_j(r(i))$. This implies that at least

$$\frac{U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^k \tilde{u}_j(r(i))}{\delta U} \tag{5}$$

tasks in $\widetilde{OPT} \cap T^{large}$ were rounded up. We next lower bound (5)

$$\begin{aligned}
&\frac{U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^k \tilde{u}_j(r(i))}{\delta U} \\
\overset{(2)}{\geq}\ & \frac{U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large})}{\delta U} + \frac{-\sum_{j=1}^k \left(d(OPT_R^{(j)} \cap T_{r(i)}) + \delta^2 U\right)}{\delta U} \\
\overset{(3)}{\geq}\ & \frac{U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large})}{\delta U} + \frac{-\delta(1+\epsilon)U - \sum_{j=1}^k d(OPT_R^{(j)} \cap T_{r(i)})}{\delta U} \\
\overset{(1)}{\geq}\ & \frac{U' - d(OPT \cap T' \cap T_{r(i)}) - \epsilon U/3 - d(\widetilde{OPT} \cap T^{large})}{\delta U} + \frac{-\delta(1+\epsilon)U - \sum_{j=1}^k d(OPT_R^{(j)} \cap T_{r(i)})}{\delta U} \\
=\ & \frac{1+\epsilon}{\delta} - \frac{\epsilon}{3\delta} - (1+\epsilon) - \frac{d(OPT \cap T' \cap T_{r(i)})}{\delta U} + \frac{d(\widetilde{OPT} \cap T^{large}) + \sum_{j=1}^k d(OPT_R^{(j)} \cap T_{r(i)})}{\delta U} \\
=\ & \frac{\epsilon}{\delta} - \frac{\epsilon}{3\delta} - (1+\epsilon) + \frac{U - d(OPT \cap T' \cap T_{r(i)})}{\delta U} + \frac{-d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^k d(OPT_R^{(j)} \cap T_{r(i)})}{\delta U} \\
\geq\ & \frac{\epsilon}{2\delta} + \frac{U - d(OPT \cap T' \cap T_{r(i)})}{\delta U} + \frac{-d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^k d(OPT_R^{(j)} \cap T_{r(i)})}{\delta U} \\
\overset{(4)}{\geq}\ & \frac{\epsilon}{2\delta} + \frac{|\widetilde{OPT} \cap T^{small}|}{\delta U}.
\end{aligned}$$

This implies that $\mathrm{disc}(\widetilde{OPT} \cap T^{small})$ is at most

$$\begin{aligned}
&|\widetilde{OPT} \cap T^{small}| \\
&\quad - \left(U' - u'(r(i)) - d(\widetilde{OPT} \cap T^{large}) - \sum_{j=1}^{k} \tilde{u}_j(r(i))\right) \\
&\leq |\widetilde{OPT} \cap T^{small}| - \left(\frac{\epsilon}{2}U + |\widetilde{OPT} \cap T^{small}|\right) = -\frac{\epsilon}{2}U
\end{aligned}$$

Hence, the edge $r(i)$ has a prefix with large negative discrepancy. □

Proof of Lemma 9. The claim follows immediately from Lemmas 11, 12, 14, and 16. □

5 ROUNDING A PLAYER'S STRATEGY

In this section we prove Lemma 6: given some strategy of the player with expected profit play (for which intuitively the edges have capacity $U' = (1+\epsilon)U$), we compute a solution with expected profit at least $(1 - O(\epsilon))$play which is feasible if all the edges have capacity $U'' = (1 + O(\epsilon))U'$.

We run the player once: let $PLAY$ be the subset of tasks selected during the game (excluding T_{LP}). We let $PLAY(e) = PLAY \cap T_e$ be the tasks in $PLAY$ using edge e. It might be that $PLAY$ is not a feasible solution (not even with resource augmentation). This happens when many more small tasks than expected are rounded down on some edge; in particular, such an edge has a prefix with large (positive) discrepancy. Therefore, we will remove some tasks from $PLAY$ that we will denote as *smearable* and *preemptable* tasks.

Definition 17. *A rounded down (small) task $i \in PLAY$ is* smearable *if $PLAY(e)$ has a prefix with large discrepancy for all $e \in P(i)$.*

Intuitively, we will show that the (random) set of smearable tasks induces a feasible fractional solution to the LP in Lemma 2, which only has a cost of at most $\epsilon w(T_{LP})$.

In order to define preemptable tasks, we need some notation. For a small task i, we let the *ancestors* $A(i)$ (resp., *descendants* $D(i)$) of i be the small[1] tasks that intersect i and appear before (resp., after) i in the order of selection:

$$A(i) := \{j \in T^{small} : P(i) \cap P(j) \neq \emptyset,\ j \prec i\}$$
$$D(i) := \{j \in T^{small} : P(i) \cap P(j) \neq \emptyset,\ j \succ i\}.$$

Notice that $j \in A(i)$ if and only if $i \in D(j)$. For each edge e we define $A(i,e) := A(i) \cap T_e$ and $D(i,e) := A(i) \cap T_e$. We sometimes interpret these subsets as order sequences according to $\prec$, so that their prefixes are well defined.

If a task i is selected in a given subgame (T', E'_L, E'_R, u'), we define $\ell_R(i)$ to be the leftmost edge of $P_R(i) = P(i) \cap E'_R$. Notice that if i is an inner task, $\ell_R(i) = \ell(i)$, otherwise $\ell_R(i) > \ell(i)$. We recall that a discrepancy is large if it is at least $\frac{\epsilon}{2}U$.

Definition 18. *A rounded down (small) task $i \in PLAY$ is* preemptable *if $D(i, r(i))$ or $D(i, \ell_R(i))$ has a prefix with large discrepancy.*

Intuitively, it is unlikely that a task i selected by the player is preemtable (due to the tasks selected after i), hence we can discard preemtable tasks with a small loss in the expected profit.

We remark that if a task is large or (small and) rounded up, then it is neither smearable nor preemptable. Observe also that a task can be smearable and preemptable at the same time. Finally, note that here we do not worry about sets of tasks having a large negative discrepancy: intuitively, in this case the rounded demand of the respective tasks is larger than their actual demand, and thus we can safely include them in our solution.

Let $PLAY^{smear}$ and $PLAY^{preem}$ be the set of smearable and preemptable tasks, resp. We define $PLAY^{fit} := PLAY \setminus (PLAY^{smear} \cup PLAY^{preem})$. We output the solution $\bar{T} := T_{LP} \cup PLAY^{fit}$. In the next two sections we will prove the following critical lemmas that imply that $\bar{T}$ fulfills the properties claimed in Lemma 6.

Lemma 19. $\mathbb{E}[w(\bar{T})] = \mathbb{E}[w(PLAY^{fit})] + w(T_{LP}) \geq (1-\epsilon)\text{play}$.

Lemma 20. *For each $e \in E$, deterministically*

$$d(\bar{T} \cap T_e) = d((PLAY^{fit} \cup T_{LP}) \cap T_e) \leq (1 + O(\epsilon))U'.$$

[1]We remark that it is possible to extend the ancestor/descendent relation to large tasks, and to tasks that do not intersect. However, this would make our proofs more complicated.

5.1 Profit of smearable and preemptable Tasks

In this subsection we prove Lemma 19. While we lose the profit of the smearable and preemptable tasks, we will show in the following that their expected profit is compensated by the profit of T_{LP} (which we get deterministically) and of $PLAY^{fit}$. To this aim, we first observe that the probability x_i^{smear} of a task i to be smearable is at most ϵ^2 due to Lemma 14. We then argue that after an edge e has attained a prefix with large discrepancy (which again happens with probability at most ϵ^2), in expectation e receives an additional load of at most U'. Therefore the fractional solution induced by the probabilities x_i^{smear} uses a total capacity of at most $O(\epsilon^2)U'$ on each edge e, implying that the profit of T_{LP} is larger by a factor $\Omega(1/\epsilon)$ than the profit of x.

Lemma 21. $\mathbb{E}[w(PLAY^{smear})] \leq O(\epsilon) \cdot w(T_{LP})$.

Proof. Let $X_i^{smear} \in \{0,1\}$ be a random variable which is 1 iff i is smearable, and let $x_i^{smear} := \mathbb{E}[X_i^{smear}] = Pr[X_i^{smear} = 1]$. One has $\mathbb{E}[w(PLAY^{smear})] = \sum_i w(i)x_i^{smear} = \sum_{i \in T^{small}} w(i)x_i^{smear}$. Let $\mathcal{E}_e$ denote the event that some prefix of $PLAY(e)$ has large discrepancy. For any $i \in T^{small}$ and an arbitrary fixed $f \in P(i)$,

$$x_i^{smear} = Pr[X_i^{smear} = 1]$$
$$= Pr[i \in PLAY \wedge \forall e \in P(i) : \mathcal{E}_e] \leq Pr[\mathcal{E}_f] \overset{\text{Lem. 14}}{\leq} \epsilon^2.$$

We will next show that for each edge e one has

$$\sum_{i \in T^{small} \cap T_e} x_i^{smear} \leq 3\epsilon^2 U'. \tag{6}$$

This implies that scaling up the variables x_i^{smear} by a factor $\frac{1}{3\epsilon}$ one obtains a feasible solution for the LP defining T_{LP}. As a consequence

$$\mathbb{E}[w(PLAY^{smear})] = \sum_{i \in T^{small}} w(i)x_i^{smear} \leq 3\epsilon\, w(T_{LP}).$$

It remains to prove (6) for a given edge e. Let $PLAY_1(e) \subseteq PLAY \cap T_e$ denote the largest prefix of the small tasks in $PLAY \cap T_e$ that does *not* have a large discrepancy, and let $PLAY_2(e) := (PLAY \cap T_e) \setminus PLAY_1(e)$. Observe that deterministically $|PLAY_1(e)| \leq U' + \epsilon U/2$ since otherwise the discrepancy of $PLAY_1(e)$ must be large. Also, $\sum_{i \in T^{small} \cap T_e} \mathbb{E}[|PLAY_2(e)||\mathcal{E}_e] \leq U'$ since after $\mathcal{E}_e$ happens, in expectation the player can select at most U' more small tasks using e. Thus, we have that

$$\sum_{i \in T^{small} \cap T_e} x_i^{smear} = \sum_{i \in T^{small} \cap T_e} \mathbb{E}[X_i^{smear}]$$
$$\leq Pr[\mathcal{E}_e] \cdot (\mathbb{E}[|PLAY_1(e)||\mathcal{E}_e] + \mathbb{E}[|PLAY_2(e)||\mathcal{E}_e])$$
$$\leq \epsilon^2 \cdot (U' + \epsilon U/2 + U') \leq 3\epsilon^2 U'. \qquad \square$$

If a task i is preemptable, then after i was selected either $D(i, r(i))$ or $D(i, \ell_R(i))$ attains a prefix with large discrepancy. This is unlikely, due to the following analogue of Lemma 14.

Lemma 22. *Fix a strategy of the player and condition on $i \in PLAY$. Then for any $e \in P(i)$, $D(i,e)$ has a prefix with large or large negative discrepancy with probability at most ϵ^2.*

Proof. The proof is identical to the proof of Lemma 14, with the difference that here S is the sequence $D(i,e)$. $\square$

Lemma 23. $\mathbb{E}[w(PLAY^{preem})] \le \epsilon\, \mathbb{E}[w(PLAY)]$.

Proof. Let x_i^{preem} be the probability of task i to be preemptable. One has

$$\mathbb{E}[w(PLAY^{preem})] = \sum_i w(i) x_i^{preem} = \sum_{i \in T^{small}} w(i) x_i^{preem}.$$

We fix a small task $i \in PLAY$ and define $\mathcal{D}_e$ to be the event that $D(i,e)$ has a prefix of large discrepancy. Then

$$\begin{aligned} x_i^{preem} &= Pr[i \in PLAY] \cdot Pr[\mathcal{D}_{r(i)} \vee \mathcal{D}_{\ell_R(i)} | i \in PLAY] \\ &\le Pr[\mathcal{D}_{r(i)} | i \in PLAY] + Pr[\mathcal{D}_{\ell_R(i)} | i \in PLAY] \\ &\overset{\text{Lem. 22}}{\le} 2\epsilon^2 \le \epsilon. \end{aligned}$$

The claim follows. □

Proof of Lemma 19. One has

$$\begin{aligned} & w(T_{LP}) + \mathbb{E}[w(PLAY^{fit})] \\ &= w(T_{LP}) + \mathbb{E}[w(PLAY)] - \mathbb{E}[w(PLAY^{smear} \cup PLAY^{preem})] \\ &\overset{\text{Lem. 21 and 23}}{\ge} (1 - O(\epsilon)) w(T_{LP}) + (1-\epsilon)\mathbb{E}[w(PLAY)] \\ &\ge (1 - O(\epsilon))\text{play}. \end{aligned}$$

□

5.2 Demand of $PLAY^{fit}$

In this subsection we prove Lemma 20. First we need some preliminaries. Recall that a task j is boundary if in some subgame (T', E'_L, E'_R, u') we have that $j \in T'$, some task i is selected such that i is large or rounded up, and then j is included in the set T'_R of tasks in the right subgame due to the fact that j intersects both E'_{RL} and E'_{RR}. In this case we say that *task i made j boundary*, and we set $bn(j) = i$. We define $BN(j)$ to be all tasks (including j itself) that are made boundary by i. Notice that by construction $\ell_R(k) = \ell_R(j)$ for each $k \in BN(j)$. For each inner task j' we define $BN(j') := \emptyset$ and let $bn(j')$ undefined.

The following Lemma is similar in spirit to Lemma 15, but better specifies some properties that we will need later.

Lemma 24. *For $i \in PLAY$ and $j \in A(i) \setminus BN(i)$, one of the following properties holds:*

1. *i is boundary, $j \ge bn(i)$, and $r(j) < \ell_R(i)$ (hence $P(j) \cap P_R(i) = \emptyset$);*
2. *$\ell(i) \in P_R(j)$ and, if i is boundary, $j \prec bn(i)$. Furthermore, $P(i) \subseteq P_R(j)$ or j is rounded down*

Proof. If $j = bn(i)$ then 1. trivially holds, hence assume next that $j \neq bn(i)$. Suppose that for some subgame (T', E'_L, E'_R, u') one has $i, j \in T'$, and then j is assigned to a left subgame $(T'_L, E'_L, E'_{RL}, u'_L)$ while i is assigned to a right subgame $(T'_R, E'_{RL}, E'_{RR}, u'_R)$ (the vice versa is not possible since $j \in A(i)$). Remember that this is due to the selection of some large or rounded up task k. In this case, since $P(j) \subseteq E'_L \cup E'_{RL}$, $P(i) \subseteq E'_{RL} \cup E'_{RR}$ and $P(j) \cap P(i) \neq \emptyset$, one has that $P(i) \cap E'_{RL} \neq \emptyset$. Hence in particular i is a boundary task with $bn(i) = k$. Since $P_R(i) = P(i) \cap E'_{RR}$, one has $P(j) \cap P_R(i) = \emptyset$. Furthermore, $j \succ k = bn(i)$.

The only other possibility is that for some subgame (T', E'_L, E'_R, u') one has $i, j \in T'$ and then $j \neq bn(i)$ is selected in this subgame. Assume by contradiction that i is already boundary at this point, hence $bn(i) \prec j$. By the order in which tasks in T' are considered, this implies that also j is boundary. In particular, $\ell_R(j) = \ell_R(i) = \ell(E'_R)$. This however implies $j \in BN(i)$, which is excluded by assumption. Therefore one has that $j \prec bn(i)$ if i is boundary. Furthermore we have $P(i) \subseteq E'_R$. If j is boundary, then necessarily $\ell_R(j) = \ell(E'_R) \le \ell(i)$. Otherwise j is the task in T' with the leftmost left edge $\ell(j)$, implying $\ell(j) = \ell_R(j) < \ell(i)$. In both cases this implies $\ell(i) \in P_R(j)$.

To conclude the proof, assume by contradiction that $P(i) \not\subseteq P_R(j)$ and j is rounded up. In this case j would make i boundary, i.e. $bn(i) = j$. This was excluded by the previous cases. □

Corollary 25. *For $i \in PLAY$, let $A'(i) := A(i)$ if i is inner and $A'(i) := \{j \in A(i) : j \prec bn(i)\}$ otherwise. Then for any $A'' \subseteq A'(i)$, $disc(A'', e)$ is a non-increasing function of $e \in P(i)$ going from left to right.*

Proof. Consider the value of $disc(\{j\}, e)$ for $e \in P(i)$ and $j \in A'(i)$. It is sufficient to show that this quantity is non-decreasing for e going from left to right. Lemma 24 implies that case (2) holds. If $P_R(j)$ spans $P(i)$, the discrepancy of j is uniform over $P(i)$. Otherwise it must be the case that j is rounded down and $P_R(j)$ intersects some left subpath of $P(i)$. In this case $disc(\{j\}, e)$ for $e \in P(i)$ going from left to right is initially 1 and then becomes 0. The claim follows. □

Lemma 20 follows easily from the next lemma.

Lemma 26. *For every $e \in E$, $disc(PLAY^{fit}, e) \le 2\epsilon\, U$.*

Proof of Lemma 20. It is not hard to see that, by construction, the rounded demand $d_{round}(PLAY, e)$ of $PLAY$ (hence of $PLAY^{fit}$) on e is (deterministically) at most U'. Then,

$$\begin{aligned} d(\bar{T} \cap T_e) &= d(T_{LP} \cap T_e) + d(PLAY^{fit} \cap T_e) \\ &\le \epsilon U + d(PLAY^{fit} \cap T_e) \\ &= \epsilon U + d_{round}(PLAY^{fit}, e) + disc(PLAY^{fit}, e) \\ &\overset{\text{Lem. 26}}{\le} \epsilon U + U' + 2\epsilon U. \end{aligned}$$ □

We prove Lemma 26 by contradiction. Let us imagine to add the tasks of $PLAY^{fit}$ one by one to a solution $PLAY'$ in their order of selection until the condition $disc(PLAY', e) \le 2\epsilon\, U$ is satisfied for every edge. Let $i_0 \in PLAY^{fit}$ be the first task which violates this condition, namely

$$disc(PLAY' \cup \{i_0\}, f) = disc(A(i_0) \cap PLAY^{fit} \cup \{i_0\}) > 2\epsilon\, U$$

for some edge f. Notice that i_0 must be a (small) rounded down task since for the remaining tasks the discrepancy is always non-positive. Observe also that $f \in P(i_0)$. More precisely, $f \in P_R(i_0)$ since the discrepancy of i_0 is zero on the remaining edges. Next lemma shows that $A(i_0) \cap PLAY^{fit}$ must have large enough discrepancy on $\ell_R(i_0)$.

Lemma 27. *We have that $disc(A(i_0) \cap PLAY^{fit}, \ell_R(i_0)) \ge 2\epsilon U$.*

Proof. We already argued that $disc(A(i_0)\cap PLAY^{fit}\cup\{i_0\}, f) > 2\epsilon U$ for some $f \in P_R(i_0)$, hence $disc(A(i_0) \cap PLAY^{fit}, f) \geq 2\epsilon U$. Let us show that $disc(A(i_0) \cap PLAY^{fit}, \ell_R(i_0)) \geq disc(A(i_0) \cap PLAY^{fit}, f)$. Notice that for $j \in BN(i_0)$ by construction $P_R(i_0) \subseteq P_R(j)$, hence $disc(\{j\}, e)$ is uniform over $e \in P_R(i_0)$. By Lemma 24, the only tasks $j \in A(i_0) \setminus BN(i_0)$ that provide a non uniform contribution to the discrepancy over $P_R(i_0)$ (namely the tasks j whose $P_R(j)$ intersect $P_R(i_0)$ without spanning it), are rounded down tasks with $\ell(i_0) \in P_R(j)$. For these tasks j, $disc(\{j\}, e)$ is non-decreasing for $e \in P(i_0)$ going from left to right. The claim follows. □

We next partition $A(i_0, \ell_R(i_0))\cap PLAY^{fit}$ into a few subsets. Our goal is to show that each such set must have a small enough discrepancy on $\ell_R(i_0)$, hence contradicting Lemma 27. Define $A_{BN} := A(i_0, \ell_R(i_0)) \cap PLAY^{fit} \cap BN(i_0)$ and $A_{\overline{BN}} := A(i_0, \ell_R(i_0)) \cap PLAY^{fit} \setminus BN(i_0)$. The next lemma implies that we select the tasks in $A_{\overline{BN}}$ before the tasks in A_{BN}.

Lemma 28. *Let $i \in PLAY$ be a boundary task, $i' \in A(i) \setminus BN(i)$ and $i'' \in A(i) \cap BN(i)$. Then $i' \prec i''$.*

Proof. By definition $i'' \succ bn(i)$, hence the claim follows if $i' \preceq bn(i)$. Suppose then that $i' \succ bn(i)$. In this case at the time of the selection of $bn(i)$ in some subgame (T', E'_L, E'_R, u'), the player creates a left subgame $(T'_L, E'_L, E'_{RL}, u'_L)$ and a right subgame $(T'_R, E'_{RL}, E'_{RR}, u'_R)$. Notice that $i'' \in T'_R$. Furthermore, since $i' \notin BN(i)$, one has $P(i') \subseteq E'_L \cup E'_{RL}$ and hence $i' \in T'_L$. The claim follows since the left subgame is played first. □

The following lemma implies an upper bound on

$$disc(A_{BN}, \ell_R(i_0)).$$

Remark 1. *For every $A \subseteq PLAY$ and every edge $e \in E$, $disc(A, e) \geq disc(A \cap PLAY^{fit}, e)$. This comes from the fact that the only tasks in $PLAY$ with negative discrepancy on some edges are small rounded up tasks, and they belong to $PLAY^{fit}$ by definition.*

Lemma 29. *For any boundary task $i \in PLAY$ and any $e \in P_R(i)$, $disc(A(i) \cap BN(i) \cap PLAY^{fit}, e) \leq \epsilon U/2$.*

Proof. Assume by contradiction that

$$disc(A(i) \cap BN(i) \cap PLAY^{fit}, e) \geq \epsilon U/2 + 1.$$

For each $j \in A(i) \cap BN(i)$, by construction $P_R(i) \subseteq P_R(j)$, hence

$$\begin{aligned} &disc(A(i) \cap BN(i) \cap PLAY^{fit}, \ell_R(i)) \\ &\quad = disc(A(i) \cap BN(i) \cap PLAY^{fit}, e) \geq \epsilon U/2 + 1. \end{aligned}$$

Consider the task $j^* \in A(i, \ell_R(i)) \cap BN(i) \cap PLAY^{fit}$ which is selected first. We next show that $D(j^*, \ell_R(i))$ has a prefix with large discrepancy on $\ell_R(i) = \ell_R(j^*)$ and hence j^* is preemptable, which contradicts $j^* \in PLAY^{fit}$. By Lemma 28, since $j^* \in BN(i)$, we have that $D' := D(j^*, \ell_R(i)) \cap A(i)$, which is a prefix of $D(j^*, \ell_R(i))$, does not contain any task $i' \notin BN(i)$. Thus, $D' = A(i, \ell_R(i))\cap BN(i)\setminus\{j^*\}$. Using Remark 1, we can conclude that

$$\begin{aligned} disc(D', \ell_R(i)) &\geq disc(A(i) \cap BN(i), \ell_R(i)) - 1 \\ &\geq disc(A(i) \cap BN(i) \cap PLAY^{fit}, \ell_R(i)) - 1 \geq \epsilon U/2. \end{aligned}$$

□

Lemma 30. $disc(A_{BN}, \ell_R(i_0)) \leq \epsilon U/2$.

Proof. The claim is trivial if i_0 is inner (since $BN(i_0) = \emptyset$) and follows from Lemma 29 otherwise. □

We analyze now the discrepancy of $A_{\overline{BN}}$ on $\ell_R(i_0)$, i.e., the value $disc(A_{\overline{BN}}, \ell_R(i_0))$. If $A_{\overline{BN}} = \emptyset$ then $disc(A_{\overline{BN}}, \ell_R(i_0)) = 0$. Assume now that $A_{\overline{BN}} \neq \emptyset$. We identify a special task $i^* \in A_{\overline{BN}}$ which we define to be the task $i \in A_{\overline{BN}}$ with the leftmost right edge $r(i)$. Assume first that $r(i^*) > r(i)$.

Lemma 31. *If $A_{\overline{BN}} \neq \emptyset$ and $r(i^*) > r(i_0)$, then $disc(A_{\overline{BN}}, \ell_R(i_0)) < \epsilon U/2$.*

Proof. Assume by contradiction that $disc(A_{\overline{BN}}, \ell_R(i_0)) \geq \frac{\epsilon U}{2}$. Let us show that i_0 is smearable, which contradicts $i_0 \in PLAY^{fit}$. We already argued before that i_0 is rounded down, hence it is sufficient to show that $PLAY(f)$ has a prefix with large discrepancy for each $f \in P(i_0)$. Define $A'(i_0)$ as in Corollary 25. Notice that $A'(i_0) \cap T_f$ is a prefix of $PLAY(f)$. We know that $disc(A'(i_0), e)$ is non-decreasing over $e \in P(i_0)$ going from left to right. Observe also that $A'(i_0) \cap PLAY^{fit} \cap T_{\ell_R(i_0)} = A_{\overline{BN}}$: this is obvious if i_0 is inner and otherwise it follows from case (2) of Lemma 24. Hence for $f \leq \ell_R(i_0)$,

$$\begin{aligned} disc(A'(i_0), f) &\geq disc(A'(i_0), \ell_R(i_0)) \\ &\overset{\text{Rem. 1}}{\geq} disc(A'(i_0) \cap PLAY^{fit}, \ell_R(i_0)) \\ &= disc(A_{\overline{BN}}, \ell_R(i_0)) \geq \epsilon U/2. \end{aligned}$$

Consider next $f > \ell_R(i_0)$ (namely, f is to the right of $\ell_R(i_0)$). From the above inequalities it is sufficient to show that $disc(A'(i_0) \cap PLAY^{fit}, f) = disc(A'(i_0)\cap PLAY^{fit}, \ell_R(i_0))$. From Lemma 24, the only possibility for this equality not to hold is that some rounded down task $j \in A'(i_0) \cap PLAY^{fit} \cap T_{\ell_R(i_0)} = A_{\overline{BN}}$ ends before f, i.e. $r(j) < f$. However this is impossible by the definition of i^* since $j \in A_{\overline{BN}}$, hence $r(j) > r(i^*) > r(i_0) \geq f$. □

Suppose now that $r(i^*) < r(i)$ (we cannot have $r(i^*) = r(i)$ since the end vertices of the tasks are pairwise distinct by assumption). We partition $A_{\overline{BN}}$ into three sets: we define $A_1 := A_{\overline{BN}} \cap A(i^*) \setminus BN(i^*)$, $A_2 := A_{\overline{BN}} \cap A(i^*) \cap BN(i^*)$, and $A_3 := A_{\overline{BN}} \setminus A(i^*)$. We show for each of them separately that it has a small discepancy on $\ell_R(i_0)$.

For A_1 we use a proof similar in spirit to the one of Lemma 31.

Lemma 32. *If $A_{\overline{BN}} \neq \emptyset$ and $r(i^*) < r(i_0)$, then $disc(A_1, \ell_R(i_0)) < \epsilon U/2$.*

Proof. Assume by contradiction that $disc(A_1, \ell_R(i_0)) \geq \frac{\epsilon U}{2}$. Let us show that i^* is smearable, which contradicts $i^* \in PLAY^{fit}$. By Lemma 24, since $\ell_R(i_0) \in P(i^*)$, we are in case (2). More precisely, since $r(i^*) < r(i_0)$ and hence $P(i_0) \not\subseteq P_R(i^*)$, one has that i^* is rounded down (and $i^* \prec bn(i_0)$ if i_0 is boundary).

Therefore it is sufficient to show that $PLAY(f)$ has a prefix with large discrepancy for each $f \in P(i^*)$. We observe that $\ell_R(i^*) \leq \ell_R(i_0)$. Indeed, suppose by contradiction that $\ell_R(i_0) < \ell_R(i^*)$, hence $\ell(i_0) \leq \ell_R(i_0) < \ell_R(i^*) \leq r(i^*) < r(i_0)$. Then i^* or an earlier task declares i_0 as boundary. This is impossible if i_0 is inner and otherwise because $bn(i_0) \preceq i^* \prec bn(i_0)$.

Let $A'(i^*)$ be defined as in Corollary 25. Observe that $A'(i^*) \cap T_f$ is a prefix of $PLAY(f)$. We know that $disc(A'(i^*), e)$ is a non-decreasing function over $e \in P(i^*)$ going from left to right. Observe also that $A_1 = A'(i^*) \cap T_{\ell_R(i_0)} \cap PLAY^{fit}$. To see that, notice that $A_1 = A(i^*, \ell_R(i_0)) \setminus BN(i^*) \cap PLAY^{fit}$. The claim follows if i^* is inner. Otherwise, by Lemma 24 any $j \in A(i^*) \setminus A'(i^*)$ (which satisfies $j \geq bn(i^*)$) must have $r(j) < \ell_R(i^*) \leq \ell_R(i_0)$, thus j is not contained in $A(i^*, \ell_R(i_0))$. Hence for any $f \leq \ell_R(i_0)$,

$$disc(A'(i^*), f) \geq disc(A'(i^*), \ell_R(i_0))$$
$$\overset{\text{Rem. 1}}{\geq} disc(A'(i^*) \cap PLAY^{fit}, \ell_R(i_0)) = disc(A_1, \ell_R(i_0)) \geq \epsilon U/2.$$

Assume next $f > \ell_R(i_0)$, hence $f \in P_R(i_0) \cap P(i^*)$. Let us show that $disc(A'(i^*), f) = disc(A'(i^*), \ell_R(i_0))$. Recall that $\ell_R(i^*) \leq \ell_R(i_0)$. Then, by the same argument as in the proof of Lemma 31, we have $disc(A'(i^*), f) = disc(A'(i^*), \ell_R(i^*))$. Together with the fact that $disc(A'(i^*), e)$ is non-decreasing over $P(i^*)$ this implies $disc(A'(i^*), f) = disc(A'(i^*), \ell_R(i_0))$ as desired. □

We can bound the discrepancy of A_2 on $\ell_R(i_0)$ with Lemma 29.

Lemma 33. *If $A_{\overline{BN}} \neq \emptyset$ and $r(i^*) < r(i_0)$, we have that $disc(A_2, \ell_R(i_0)) \leq \epsilon U/2$.*

Proof. Notice that $A_2 = A_{\overline{BN}} \cap A(i^*) \cap BN(i^*) = A(i^*) \cap BN(i^*) \cap T_{\ell_R(i_0)}$. Hence

$$disc(A_2, \ell_R(i_0)) = disc(A(i^*) \cap BN(i^*), \ell_R(i_0)).$$

As already observed in the proof of Lemma 32, $\ell_R(i^*) \leq \ell_R(i_0)$, hence $\ell_R(i_0) \in P_R(i^*)$. The claim follows from Lemma 29. □

We can argue that, if the discrepancy of A_3 is too large on $\ell_R(i_0)$, then i^* is preemptable, leading to a contradiction.

Lemma 34. *If $A_{\overline{BN}} \neq \emptyset$ and $r(i^*) < r(i_0)$, we have that $disc(A_3, \ell_R(i_0)) \leq \epsilon U/2$.*

Proof. Assume by contradiction that $disc(A_3, \ell_R(i_0)) \geq \frac{\epsilon U}{2} + 1$. Let us show that i^* is preemptable, contradicting $i^* \in PLAY^{fit}$. We already argued in the proof of Lemma 32 that i^* is rounded down. Let $D' = \{j \in D(i^*, r(i^*)) : j \prec i_0\}$ if i_0 is inner and $D' = \{j \in D(i^*, r(i^*)) : j \prec bn(i_0)\}$ otherwise. Notice that D' is a prefix of $D(i^*, r(i^*))$. It is sufficient to show that $disc(D', r(i^*))$ is large.

Let us first show that $A_3 \cap T_{r(i^*)} \setminus \{i^*\} = D' \cap PLAY^{fit}$. We start by observing that $A_3 \cap T_{r(i^*)} \setminus \{i^*\} \subseteq D'$. Indeed consider any $j \in A_3 \cap T_{r(i^*)} \setminus \{i^*\}$: $j \in PLAY^{fit}$, j contains $r(i^*)$ (by the definition of i^*) and j is an ancestor of i_0. The claim follows if i_0 is inner. Otherwise it is sufficient to show that $j \prec bn(i_0)$: this follows from Lemma 24 since j contains $\ell_R(i_0)$ (hence in particular intersects $P_R(i_0)$). Next assume by contradiction that there exists $j \in D' \cap PLAY^{fit} \setminus (A_3 \cap T_{r(i^*)} \setminus \{i^*\})$. Then j would be an ancestor of i_0 starting to the right of $\ell_R(i_0)$: this is excluded by Lemma 24.

By the same lemma, $A_3 \subseteq A'(i_0)$ with $A'(i_0)$ defined as in Corollary 25. Then $disc(A_3, \ell_R(i_0)) \geq disc(A_3, r(i^*))$. Let us show that equality holds. Indeed otherwise, by Lemma 24, there must exist a rounded down task $j \in A_3$ which ends before $r(i^*)$. However this would contradict the definition of i^* since $j \in A_{\overline{BN}}$. We can conclude that

$$disc(D', r(i^*)) \overset{\text{Rem. 1}}{\geq} disc(D' \cap PLAY^{fit}, r(i^*))$$
$$= disc(A_3 \setminus \{i^*\}, r(i^*)) \geq disc(A_3, r(i^*)) - 1$$
$$= disc(A_3, \ell_R(i_0)) - 1 \geq \epsilon U/2. \quad \square$$

Proof of Lemma 26. By Lemma 30, $disc(A_{BN}, \ell_R(i_0)) \leq \frac{\epsilon U}{2}$. If $A_{\overline{BN}} = \emptyset$ one has $disc(A_{\overline{BN}}, \ell_R(i_0)) = 0$. Otherwise, if $r(i^*) > r(i)$, then $disc(A_{\overline{BN}}, \ell_R(i_0)) < \frac{\epsilon U}{2}$ by Lemma 31. If $r(i^*) < r(i)$, from Lemmas 32, 33, and 34 we conclude that

$$disc(A_{\overline{BN}}, \ell_R(i_0)) = disc(A_1, \ell_R(i_0)) + disc(A_2, \ell_R(i_0)) + disc(A_3, \ell_R(i_0)) \leq 3\epsilon U/2 - 1.$$

In both cases we get a contradiction since

$$2\epsilon U \overset{\text{Lem. 27}}{\leq} disc(A(i_0) \cap PLAY^{fit}, \ell_R(i_0))$$
$$= disc(A_{\overline{BN}}, \ell_R(i_0)) + disc(A_{BN}, \ell_R(i_0)) \leq 2\epsilon U - 1. \quad \square$$

6 THE GENERAL CASE

We now study UFP in the general case, so in particular with arbitrary edge capacities and without resource augmentation. Due to space constraints, we can only sketch the basic ideas here, postponing the details to a journal version. We remark that also in prior work, e.g., [27–30], first an algorithm for the case of (almost) uniform capacities and resource augmentation was constructed, which was then extended to the general case (without resource augmentation). The critical idea is to use a slack lemma originally introduced in [11], and later refined in [27–30]. Roughly speaking, there exists a nearly optimal solution OPT' that leaves some amount of slack on each edge (possibly 0). The slack values can be classified in different levels $0 = sl(0) < \ldots < sl(q)$. Then intuitively on an edge e with slack $sl(\ell)$ (of level ℓ) there are only a constant number of tasks in OPT' with demand $\Omega(sl(\ell))$ that use e. The remaining tasks of OPT' that use e have total demand $O(\epsilon \cdot sl(\ell + 1))$. In prior work [27–30], the slack level of each edge was *guessed* within a dynamic program proceeding level by level. We can rephrase the same approach in the form of a game. In more details, the idea is to design a game that is played in different slack levels, i.e., each subgame is associated with one slack level ℓ. In such a subgame, the player needs to leave essentially $sl(\ell)$ units of slack on each edge. Since also OPT' does this, we can still show that there is a strategy for the player that achieves essentially the same profit as OPT'.

Given such a subgame of some slack level ℓ, besides selecting and discarding tasks as usual, the player also defines subpaths with slack level at least $\ell + 1$ (in a similar way as the DPs in prior work guessed such subpaths). Whenever a subpath E' of the latter type is defined, the game splits into *three* subgames: left and right subgames as before, still of level ℓ, and a middle subgame of level $\ell + 1$ on subpath E'. The *interface* between these subgames is captured by demand profiles similarly to the uniform capacity case, plus a constant number of tasks with relatively large demand w.r.t. $sl(\ell)$. Combining all the possible cases together requires a relatively straightforward, but also rather technical, definition of how this game is played and into which subgames we recurse, depending on the choices of the player. Other than that, the rest of the analysis follows quite closely the analysis in the uniform case (with resource augmentation). In particular, the role of the extra capacity given by resource augmentation is replaced now by the slack capacity left by OPT' on each edge. This allows one to analyze the behaviour of the player in each level ℓ (and to round the corresponding strategy) in a fairly independent way. There is one more delicate technical complication due to the fact that, in the rounding stage, certain tasks might be selected multiple times at different levels. We resolve this issue by computing a proper LP solution at each level (rather than just the single set T_{LP} as above), and then using a charging argument similar in spirit to the one used in [30] to resolved a similar issue (however the approach in [30] leads to a loss of a substantial fraction of the profit, while here we lose only an ϵ fraction of it).

REFERENCES

[1] Anna Adamaszek, Parinya Chalermsook, Alina Ene, and Andreas Wiese. 2016. Submodular Unsplittable Flow on Trees. In *IPCO (Lecture Notes in Computer Science, Vol. 9682)*. 337–349. https://doi.org/10.1007/978-3-319-33461-5_28

[2] Anna Adamaszek and Andreas Wiese. 2013. Approximation schemes for maximum weight independent set of rectangles. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 400–409.

[3] Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. 2013. Constant Integrality Gap LP Formulations of Unsplittable Flow on a Path. In *IPCO*. 25–36. https://doi.org/10.1007/978-3-642-36694-9_3

[4] Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. 2014. A Mazing 2+ε Approximation for Unsplittable Flow on a Path. In *SODA*. 26–41.

[5] N. Bansal, A. Chakrabarti, A. Epstein, and B. Schieber. 2006. A quasi-PTAS for unsplittable flow on line graphs. In *STOC*. ACM, 721–729.

[6] N. Bansal, Z. Friggstad, R. Khandekar, and R. Salavatipour. 2009. A logarithmic approximation for unsplittable flow on line graphs. In *SODA*. 702–709.

[7] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber. 2000. A unified approach to approximating resource allocation and scheduling. In *STOC*. 735–744.

[8] R. Bar-Yehuda, M. Beder, Y. Cohen, and D. Rawitz. 2006. Resource Allocation in Bounded Degree Trees. In *ESA*. 64–75.

[9] Reuven Bar-Yehuda, Michael Beder, Yuval Cohen, and Dror Rawitz. 2009. Resource allocation in bounded degree trees. *Algorithmica* 54, 1 (2009), 89–106.

[10] Reuven Bar-Yehuda, Michael Beder, and Dror Rawitz. 2017. A Constant Factor Approximation Algorithm for the Storage Allocation Problem. *Algorithmica* 77, 4 (2017), 1105–1127. https://doi.org/10.1007/s00453-016-0137-8

[11] Jatin Batra, Naveen Garg, Amit Kumar, Tobias Mömke, and Andreas Wiese. 2015. New Approximation Schemes for Unsplittable Flow on a Path. In *SODA*. 47–58. https://doi.org/10.1137/1.9781611973730.5 arXiv:http://epubs.siam.org/doi/pdf/10.1137/1.9781611973730.5

[12] Paul Bonsma, Jens Schulz, and Andreas Wiese. 2014. A Constant-Factor Approximation Algorithm for Unsplittable Flow on Paths. *SIAM J. Comput.* 43 (2014), 767–799.

[13] Gruia Călinescu, Amit Chakrabarti, Howard J. Karloff, and Yuval Rabani. 2011. An improved approximation algorithm for resource allocation. *ACM Transactions on Algorithms* 7, Article 48 (2011), 48:1–48:7 pages. https://doi.org/10.1145/2000807.2000816

[14] Venkatesan T. Chakaravarthy, Anamitra R. Choudhury, Shalmoli Gupta, Sambuddha Roy, and Yogish Sabharwal. 2014. Improved Algorithms for Resource Allocation under Varying Capacity. In *ESA*. 222–234.

[15] A. Chakrabarti, C. Chekuri, A. Gupta, and A. Kumar. 2007. Approximation Algorithms for the Unsplittable Flow Problem. *Algorithmica* 47 (2007), 53–78.

[16] Parinya Chalermsook. 2011. Coloring and maximum independent set of rectangles. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (2011), 123–134.

[17] P. Chalermsook and J. Chuzhoy. 2009. Maximum independent set of rectangles. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09)*. SIAM, 892–901.

[18] C. Chekuri, A. Ene, and N. Korula. 2009. Unsplittable Flow in Paths and Trees and Column-Restricted Packing Integer Programs. In *APPROX-RANDOM*. 42–55.

[19] C. Chekuri, M. Mydlarz, and F. Shepherd. 2007. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms* 3 (2007).

[20] B. Chen, R. Hassin, and M. Tzur. 2002. Allocation of bandwidth and storage. *IIE Transactions* 34 (2002), 501–507.

[21] M. Chrobak, G. Woeginger, K. Makino, and H. Xu. 2010. Caching Is Hard, Even in the Fault Model. In *ESA*. 195–206.

[22] Julia Chuzhoy and Alina Ene. 2016. On Approximating Maximum Independent Set of Rectangles. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, Irit Dinur (Ed.). IEEE Computer Society, 820–829. https://doi.org/10.1109/FOCS.2016.92

[23] A. Darmann, U. Pferschy, and J. Schauer. 2010. Resource allocation with time intervals. *Theoretical Computer Science* 411 (2010), 4217–4234. Issue 49.

[24] Zachary Friggstad and Zhihan Gao. 2015. On Linear Programming Relaxations for Unsplittable Flow in Trees. In *APPROX-RANDOM (LIPIcs, Vol. 40)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 265–283.

[25] Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. 2021. A (2+ϵ)-Approximation Algorithm for Maximum Independent Set of Rectangles. *CoRR* abs/2106.00623 (2021). arXiv:2106.00623 https://arxiv.org/abs/2106.00623

[26] Fabrizio Grandoni, Salvatore Ingala, and Sumedha Uniyal. 2015. Improved Approximation Algorithms for Unsplittable Flow on a Path with Time Windows. In *WAOA*. 13–24.

[27] Fabrizio Grandoni, Tobias Mömke, and Andreas Wiese. 2021. Faster (1+ϵ)-Approximation for Unsplittable Flow on a Path via Resource Augmentation and Back. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference) (LIPIcs, Vol. 204)*, Petra Mutzel, Rasmus Pagh, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 49:1–49:15. https://doi.org/10.4230/LIPIcs.ESA.2021.49

[28] Fabrizio Grandoni, Tobias Mömke, and Andreas Wiese. 2022. Unsplittable Flow on a Path: The Game!. In *SODA*.

[29] Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. 2017. To Augment or Not to Augment: Solving Unsplittable Flow on a Path by Creating Slack. In *SODA*. 2411–2422.

[30] Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. 2018. A (5/3 + ϵ)-approximation for unsplittable flow on a path: placing small tasks into boxes. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 607–619. https://doi.org/10.1145/3188745.3188894

[31] Fabrizio Grandoni and Thomas Rothvoß. 2016. Pricing on Paths: A PTAS for the Highway Problem. *SIAM J. Comput.* 45, 2 (2016), 216–231. https://doi.org/10.1137/140998846

[32] Fabrizio Grandoni and Andreas Wiese. 2019. Packing Cars into Narrow Roads: PTASs for Limited Supply Highway. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany (LIPIcs, Vol. 144)*, Michael A. Bender, Ola Svensson, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 54:1–54:14. https://doi.org/10.4230/LIPIcs.ESA.2019.54

[33] S. Leonardi, A. Marchetti-Spaccamela, and A. Vitaletti. 2000. Approximation Algorithms for Bandwidth and Storage Allocation Problems under Real Time Constraints. In *FSTTCS*. 409–420.

[34] Tomás Martínez-Muñoz and Andreas Wiese. 2021. FPT and FPT-Approximation Algorithms for Unsplittable Flow on Trees. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference) (LIPIcs, Vol. 204)*, Petra Mutzel, Rasmus Pagh, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 67:1–67:15. https://doi.org/10.4230/LIPIcs.ESA.2021.67

[35] Joseph S. B. Mitchell. 2021. Approximating Maximum Independent Set for Rectangles in the Plane. *CoRR* abs/2101.00326 (2021). arXiv:2101.00326 https://arxiv.org/abs/2101.00326

[36] Tobias Mömke and Andreas Wiese. 2015. A (2+ϵ) Approximation Algorithm for the Storage Allocation Problem. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 9134)*, Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann (Eds.). Springer, 973–984. https://doi.org/10.1007/978-3-662-47672-7_79

[37] Tobias Mömke and Andreas Wiese. 2020. Breaking the Barrier of 2 for the Storage Allocation Problem. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference) (LIPIcs, Vol. 168)*, Artur Czumaj, Anuj Dawar, and Emanuela Merelli (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 86:1–86:19. https://doi.org/10.4230/LIPIcs.ICALP.2020.86

[38] C. A. Phillips, R. N. Uma, and J. Wein. 2000. Off-line admission control for general scheduling problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00)*. ACM, 879–888.

[39] Andreas Wiese. 2017. A (1+ϵ)-Approximation for Unsplittable Flow on a Path in Fixed-Parameter Running Time. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland (LIPIcs, Vol. 80)*, Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 67:1–67:13. https://doi.org/10.4230/LIPIcs.ICALP.2017.67