Knapsack Problems with Sigmoid Utilities: Approximation Algorithms via Hybrid Optimization☆

Vaibhav Srivastava[a], Francesco Bullo[b]

[a] *Mechanical and Aerospace Engineering Department, Princeton University, Princeton, NJ 08544, USA,* vaibhavs@princeton.edu
[b] *Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, CA 93106, USA,* bullo@engineering.ucsb.edu

Abstract

We study a class of non-convex optimization problems involving sigmoid functions. We show that sigmoid functions impart a combinatorial element to the optimization variables and make the global optimization computationally hard. We formulate versions of the knapsack problem, the generalized assignment problem and the bin-packing problem with sigmoid utilities. We merge approximation algorithms from discrete optimization with algorithms from continuous optimization to develop approximation algorithms for these NP-hard problems with sigmoid utilities.

Keywords: sigmoid utility, S-curve, knapsack problem, generalized assignment problem, bin-packing problem, multi-choice knapsack problem, approximation algorithms, human attention allocation

1. Introduction

With the inception of the National Robotic Initiative [2], the research in the field of human-robot interaction has burgeoned. Design of robotic partners that help human operators better interact with the automaton has received significant emphasis. In complex and information rich operations, one of the key roles for these robotic partners is to help human operators efficiently focus their attention. For instance, consider a surveillance operation that requires human operators to monitor the evidence collected by autonomous agents [3, 4]. The excessive amount of information available in such systems often results in poor decisions by human operators [5]. In this setting, the robotic partner may suggest to operators the optimal duration (attention) to be allocated to each piece of evidence. To this end, the robotic partner requires efficient attention allocation algorithms for human operators.

In this paper we study certain non-convex resource allocation problems with sigmoid utilities. Examples of sigmoid utility functions include the correctness of human decisions as a function of the decision time [6, 7, 8], the effectiveness of human-machine communication as a function of the communication rate [8], human performance in multiple target search as a function of the search time [9], advertising response as a function of the investment [10], and the expected profit in bidding as a function of the bidding amount [11]. We present versions of the knapsack problem (KP), the bin-packing problem (BPP), and the generalized assignment problem (GAP) in which each item has a sigmoid utility. If the utilities are step functions, then these problems reduce to the standard knapsack problem, the bin-packing problem, and the generalized assignment problem [12, 13], respectively. Similarly, if the utilities are concave functions, then these problems reduce to standard convex resource allocation problems [14]. We will show that with sigmoid utilities optimization problems become a hybrid of discrete and continuous optimization problems.

KPs [15, 12, 13] have been extensively studied. Considerable emphasis has been on the discrete KP [12] and KPs with concave utilities [16]. Non-convex KPs also have received a significant attention. Kameshwaran *et al.* [17] study KPs with piecewise linear utilities. Moré *et al.* [18] and Burke *et al.* [19] study KPs with convex utilities. In an early work, Ginsberg [20] studies a KP in which items have identical sigmoid utilities. Freeland *et al.* [21] discuss the implications of sigmoid functions on decision models. They present an approximation algorithm for the KP with sigmoid utilities that replaces the sigmoid functions with their concave envelopes and solves the resulting convex problem. In a recent work, Ağrali *et al.* [22] consider the KP with sigmoid utilities and show that this problem is NP-hard. They relax the problem by constructing concave envelopes of the sigmoid functions and then determine the global optimal solution using branch and bound techniques. They also develop a fully polynomial-time approximation scheme (FPTAS) for the case in which decision variables are discrete.

Attention allocation for human operators has been a topic of increased research recently. In particular, the sigmoid performance functions of the human operator serving a queue of decision-making tasks have been utilized to develop optimal attention allocation policies for the operator in [23, 24]. Bertuccelli *et al.* [25] study an optimal scheduling problem in human supervisory control. They determine a sequence in which the

☆This work has been supported in part by AFOSR MURI Award-FA9550-07-1-0528 and by ARO Award W911NF-11-1-0092. A preliminary version of this work [1] entitled ”Hybrid combinatorial optimization: Sample problems and algorithms” was presented at CDC-ECC, 2011, Orlando, Florida, USA.

tasks should be serviced so that the accumulated reward is maximized.

We study optimization problems with sigmoid utilities. In the context of resource allocation problems, we show that a sigmoid utility renders a combinatorial element to the problem, and the amount of resource allocated to the associated item under an optimal policy is either zero or more than a critical value. Thus, optimization variables have both continuous and discrete features. We exploit this interpretation of optimization variables and merge algorithms from continuous and discrete optimization to develop efficient *hybrid* algorithms.

We study versions of the KP, the GAP and the BPP in which utilities are sigmoid functions of the resource allocated. In particular, we study the following problems:

First, given a set of items, a single knapsack with a fixed amount of the resource, and the sigmoid utility of each item, determine the optimal resource allocation to each item.

Second, given a set of items, multiple knapsacks, each with a fixed amount of resource, and the sigmoid utility of each item-knapsack pair, determine the optimal assignments of items to knapsacks and the associated optimal resource allocation to each item.

Third, consider a set of items with their sigmoid utilities, and an unlimited number of bins with a fixed amount of the resource available at each bin. Determine the minimum number of bins, and a mapping of each item to some bin such that an optimal allocation in the first problem is non-zero for each item in every bin.

These problems model situations in which human operators are looking at the feeds from a camera network and are deciding whether some malicious activity is present. The first problem determines the optimal duration operators should allocate to each feed such that their overall performance are optimal. The second problem determines the optimal feed assignments to identical and independently working operators as well as the optimal duration allocation for each operator. Assuming that the operators work in an optimal fashion, the third problem determines the minimum number of operators required and feed-assignments to operators such that each operator allocates a non-zero duration to each feed.

For clarity of presentation, discussions herein address these problems in the context of human decision-making. Following up on the examples of sigmoid performance functions mentioned earlier, the solutions to these problems can also be used to determine optimal human-machine communication policies, search strategies, advertisement duration allocation, and bidding strategies.

The major contributions of this work are fourfold. First, we investigate the root-cause of combinatorial effects in optimization problems with sigmoid utilities. We show that for a sigmoid function subject to a linear penalty, the optimal allocation jumps down to zero with increasing penalty rate. This jump in the optimal allocation imparts combinatorial effects to optimization problems involving multiple sigmoid functions.

Second, we study the KP with sigmoid utilities and determine a constant factor approximation algorithm for it. Our approach relies on the above combinatorial interpretation of the sigmoid functions and utilizes a combination of approximation algorithms for the binary KP and algorithms for continuous univariate optimization.

Third, we study the GAP with sigmoid utilities. We first show that the GAP with sigmoid utilities is NP-hard. We then use a KP-based algorithm for the binary GAP to develop an equivalent algorithm for the GAP with sigmoid utilities.

Fourth and finally, we study the BPP with sigmoid utilities. We first show that the BPP with sigmoid utilities is NP-hard. We then utilize the solution of the KP with sigmoid utilities to develop a next-fit-type algorithm for the BPP with sigmoid utilities.

The remainder of the paper is organized in the following way. We highlight the root cause of combinatorial effects in optimization problems with sigmoid utilities in Section 2. We study the knapsack problem with sigmoid utilities, the generalized assignment problem with sigmoid utilities, and the bin-packing problem with sigmoid utilities in Sections 3, 4, and 5, respectively. Our conclusions are presented in Section 6.

2. Sigmoid Functions and Linear Penalties

In this section we formally define sigmoid functions, explore their connections with human decision-making, and study the maximization of a sigmoid function with a linear penalty.

2.1. *Sigmoid functions*

A Lipschitz-continuous function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$f(t) = f_{\text{cvx}}(t)\mathbf{1}(t < t^{\text{inf}}) + f_{\text{cnv}}(t)\mathbf{1}(t \geq t^{\text{inf}}),$$

where f_{cvx} and f_{cnv} are monotonically non-decreasing convex and concave functions, respectively, $\mathbf{1}(\cdot)$ is the indicator function, and t^{inf} is the inflection point. The sub-derivative of a sigmoid function is unimodal and achieves its maximum at t^{inf}. Moreover, $\lim_{t\to+\infty} \partial f(t) = 0$, where ∂f represents the sub-derivative of the function f. A typical graph of a smooth sigmoid function and its derivative is shown in Figure 1.

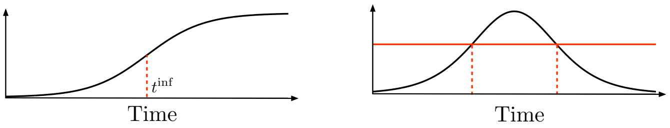

Figure 1: A typical graph of a smooth sigmoid function and its derivative.

Remark 1 (***Non-smooth sigmoid functions***)**.** For ease of presentation, we focus on smooth sigmoid functions in this paper. Our analysis extends immediately to non-smooth functions by using the sub-derivative instead of the derivative. □

Remark 2 (***Non-monotonic sigmoid functions***)**.** In several interesting budget allocation problems, e.g., [26], the sigmoid utilities are not non-decreasing functions. The algorithms proposed in this paper involve certain performance improvement heuristics that exploit the monotonicity of the utility function and hence, do not apply to problems with such sigmoid utilities. However, the proposed algorithms without the performance improvement heuristics apply to such problems, and the obtained solution is within a constant factor of the optimal. □

2.2. Sigmoid functions and human decision-making

As discussed in the introduction, sigmoid functions model the utility in several contexts. Herein, we focus on one particular context, namely, human decision-making, and detail the significance of sigmoid functions. Consider a scenario in which a human subject is exposed to a noisy stimuli for a given amount of time. Then the human subject makes a decision on the presence or absence of a signal in the stimuli. In this scenario, the probability of the human decision being correct as a function of the allocated time is modeled well by a sigmoid function. We now briefly describe some models from the human-factors and the cognitive psychology literature that suggest that a sigmoid function is an appropriate measure of the correctness of the human decision:

Pew's model: For a two-alternative forced choice task, the probability of the correct decision D_1 given that the hypothesis H_1 is true and t units of time have been spent to make the decision is:

$$\mathbb{P}(D_1|H_1,t) = \frac{p_0}{1+e^{-(at-b)}},$$

where $p_0 \in [0,1]$, $a, b \in \mathbb{R}$ are some parameters specific to the human operator [7]. Thus, according to Pew's model, the probability of the correct decision is a sigmoid function of the time spent to make the decision.

Drift diffusion model: For a two alternative forced choice task, conditioned on the hypothesis H_1, the evolution of the evidence for decision making is modeled as a drift-diffusion process [6]. That is, for a given drift rate $\beta \in \mathbb{R}_{>0}$, and a diffusion rate $\sigma \in \mathbb{R}_{>0}$, the evidence Λ at time t is normally distributed with mean βt and variance $\sigma^2 t$. The decision is made in favor of H_1 if the evidence is greater than a decision threshold $\eta \in \mathbb{R}_{>0}$. Therefore, the conditional probability of the correct decision D_1 given that the hypothesis H_1 is true and t units of time have been spent to make the decision is:

$$\mathbb{P}(D_1|H_1,t) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \int_{\eta}^{+\infty} e^{\frac{-(\Lambda-\beta t)^2}{2\sigma^2 t}} d\Lambda,$$

which is a sigmoid function of the time spent to make the decision.

Log-normal model: Reaction times of a human operator in several missions have been studied [27] and are shown to follow a log-normal distribution. In this context, a relevant performance function is the probability that the operator reacts within a given time. This corresponds to the cumulative distribution function of the log-normal distribution, which is a sigmoid function of the given time.

2.3. Maximum of a sigmoid function subject to a linear penalty

In order to gain insight into the behavior of sigmoid functions, we start with a simple problem with a very interesting result. We study the maximization of a sigmoid function subject to a linear penalty. In particular, given a sigmoid function f and a penalty rate $c \in \mathbb{R}_{>0}$, we study the following problem:

$$\underset{t\geq 0}{\text{maximize}} \quad f(t) - ct. \tag{1}$$

The derivative of a sigmoid function is not a one-to-one mapping and hence, it is not invertible. We define the pseudo-inverse of the derivative of a sigmoid function f with inflection point t^{inf}, $f^\dagger : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ by

$$f^\dagger(y) = \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = y\}, & \text{if } y \in (0, f'(t^{\text{inf}})], \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Lemma 1 (***A sigmoid function with a linear penalty***)**.** *For the optimization problem* (1)*, the optimal allocation t^* is*

$$t^* := \operatorname{argmax}\{f(\beta) - c\beta \mid \beta \in \{0, f^\dagger(c)\}\}.$$

Proof. The global maximum lies at the point where the first derivative is zero or at the boundary. The first derivative of the objective function is $f'(t) - c$. If $f'(t^{\text{inf}}) < c$, then the objective function is a decreasing function of time, and the maximum is achieved at $t^* = 0$. Otherwise, a critical point is obtained by setting the first derivative to zero. We note that $f'(t) = c$ has at most two roots. If there are two roots, then only the larger root lies in the region where the objective function is concave and hence corresponds to a maximum. Otherwise, the only root lies in the region where the objective function is concave and hence corresponds to a local maximum. The global maximum is determined by comparing the local maximum with the value of the objective function at the boundary $t = 0$. □

The optimal solution to problem (1) for different values of penalty rate c is shown in Figure 2. The optimal allocation jumps down to zero at a critical penalty rate. This jump in the optimal allocation gives rise to combinatorial effects in problems involving multiple sigmoid functions.

Definition 1 (***Critical penalty rate***)**.** For the optimization problem (1), the maximum penalty rate that yields a non-zero solution is referred to as the critical penalty rate. Formally, for a given sigmoid function f and a penalty rate $c \in \mathbb{R}_{>0}$, let the solution of the problem (1) be $t^*_{f,c}$. Then, the critical penalty rate ψ_f is defined by

$$\psi_f = \max\{c \in \mathbb{R}_{>0} \mid t^*_{f,c} \in \mathbb{R}_{>0}\}.$$

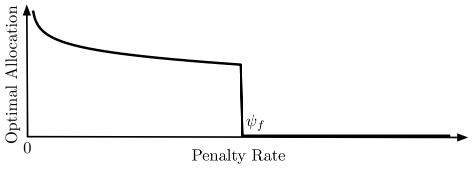

Figure 2: Optimal allocations to a sigmoid function as a function of the linear penalty

3. Knapsack Problem with Sigmoid Utilities

In this section, we consider the KP with sigmoid utilities. We first define the problem and then develop an approximation algorithm for it.

3.1. KP with Sigmoid Utilities: Problem Description

Consider a single knapsack and N items. Let the utility of item $\ell \in \{1, \dots, N\}$ be a sigmoid function $f_\ell : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Given the total available resource $T \in \mathbb{R}_{>0}$, the objective of the *KP with sigmoid utilities* is to determine the resource allocation to each item such that the total utility of the knapsack is maximized. Formally, the KP with sigmoid utilities is posed as:

$$\begin{aligned} \underset{t \succeq 0}{\text{maximize}} \quad & \sum_{\ell=1}^{N} f_\ell(t_\ell) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} t_\ell \leq T. \end{aligned} \tag{3}$$

In (3), without loss of generality, we assume that the decision variables in the resource constraint and the sigmoid utilities in the objective function are unweighted. Indeed, if the weights on the decision variables in the resource constraint are non-unity, then the weighted decision variable can be interpreted as a new scaled decision variable; while a weighted sigmoid utility is again a sigmoid utility.

The KP with sigmoid utilities models the situation in which a human operator has to perform N decision-making tasks within time T. If the performance of the human operator on task ℓ is given by the sigmoid function f_ℓ, then the optimal duration allocation to each task is determined by the solution of problem (3). We now state the following proposition from [22]:

Proposition 2 (***Hardness of the KP with sigmoid utilities***)**.** *The KP with sigmoid utilities is NP-hard, unless P = NP.*

We now present a simple example to illustrate that a naive concave relaxation of the KP with sigmoid utilities (3) may lead to an arbitrarily bad performance.

Example 1 (***Performance of a naive concave relaxation***)**.** Consider an instance of the KP with sigmoid utilities in which each sigmoid utility is identical and is defined by $f(t) = 1/(1 + \exp(-t + 5))$. Let the total available resource be $T = 8$ units and the number of items be $N = 10$. The optimal solution obtained using the procedure outlined later in the paper is to allocate the entire resource to a single item and accordingly, allocate zero resource to every other item. The value of the objective function under such an optimal policy is 0.9526.

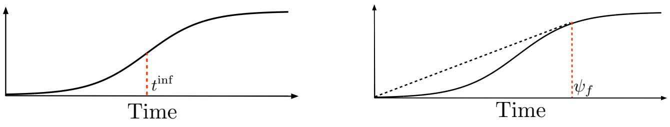

Figure 3: A sigmoid function and the associated concave envelope.

We now consider the solution to this problem obtained by a popular concave relaxation scheme. In particular, we consider the solution obtained by replacing each sigmoid function with its concave envelope (see Figure 3). An optimal solution to the resulting relaxed maximization problem is $t_\ell = T/N$, for each $\ell \in \{1, \dots, N\}$. The value of the objective function under this solution is 0.1477. Thus, the concave envelope-based policy performs badly compared to an optimal policy. In fact, the performance of the concave envelope-based policy can be made arbitrarily bad by increasing the number of items. □

Example 1 highlights that a naive concave envelope based approach may yield an arbitrarily bad performance. While such a performance can be improved using existing branch-and-bound methods [22], but in general, branch-and-bound methods may have an exponential run time. In the following, we develop an approximation algorithm for the KP with sigmoid utilities that is within a constant factor of the optimal and has a polynomial run time.

3.2. KP with Sigmoid Utilities: Approximation Algorithm

We define the Lagrangian $L : \mathbb{R}^N_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ for the knapsack problem with sigmoid utilities (3) by

$$L(\boldsymbol{t}, \alpha, \boldsymbol{\mu}) = \sum_{\ell=1}^{N} f_\ell(t_\ell) + \alpha(T - \sum_{\ell=1}^{N} t_\ell) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{t},$$

where $\alpha \in \mathbb{R}_{\geq 0}$ and $\boldsymbol{\mu} \in \mathbb{R}^N_{\geq 0}$ are Lagrange multipliers associated with the resource constraint and non-negativity constraints, respectively. Let $t_\ell^{\inf}$ be the inflection point of the sigmoid function f_ℓ and $f_\ell^\dagger$ be the pseudo-inverse of its derivative as defined in equation (2). We define the maximum value of the derivative of the sigmoid function f_ℓ by $\alpha_\ell = f'_\ell(t_\ell^{\inf})$. We also define $\alpha_{\max} = \max\{\alpha_\ell \mid \ell \in \{1, \dots, N\}\}$. We will later show that $\alpha_{\max}$ is the maximum possible value of an optimal Lagrange multiplier associated with the resource constraint.

We define the set of *inconsistent* sigmoid functions by $\mathcal{I} = \{\ell \in \{1, \dots, N\} \mid t_\ell^{\inf} > T\}$, i.e., the set of sigmoid functions for which any feasible allocation is in the convex part of the sigmoid function. Similarly and accordingly, we define the set of *consistent* sigmoid functions as $\{1, \dots, N\} \setminus \mathcal{I}$. We will show that for an inconsistent sigmoid function, the optimal allocation is either zero or T. We denote the j-th element of the standard basis of $\mathbb{R}^N$ by $\boldsymbol{e}_j$.

Since constraints in (3) are linear, the solution to (3) is *regular*, and hence the Karush-Kuhn-Tucker (KKT) conditions for optimality hold [28]. We will show that for a fixed value of the Lagrange multiplier α and consistent sigmoid functions, the KKT conditions reduce the optimization problem (3) to the α-parametrized KP defined by:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell(f_\ell^\dagger(\alpha)) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell^\dagger(\alpha) \leq T \\ & x_\ell \in \{0,1\}, \quad \forall \ell \in \{1,\dots,N\}. \end{aligned} \tag{4}$$

Define $F : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ as the optimal value of the objective function in the α-parametrized KP (4).

For a fixed value of α, (4) is a binary KP which is NP-hard. We now relax (4) to the following α-parametrized fractional KP:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell(f_\ell^\dagger(\alpha)) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell^\dagger(\alpha) \leq T \\ & x_\ell \in [0,1], \quad \forall \ell \in \{1,\dots,N\}. \end{aligned} \tag{5}$$

Define $F_{\mathrm{LP}} : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ as the optimal value of the objective function in the α-parametrized fractional KP (5). For a given α, the solution to problem (5) is obtained in the following way:

(i). sort tasks such that

$$\frac{f_1(f_1^\dagger(\alpha))}{f_1^\dagger(\alpha)} \geq \frac{f_2(f_2^\dagger(\alpha))}{f_2^\dagger(\alpha)} \geq \dots \geq \frac{f_N(f_N^\dagger(\alpha))}{f_N^\dagger(\alpha)};$$

(ii). find $k := \min\{j \in \{1,\dots,N\} \mid \sum_{i=1}^{j} f_i^\dagger(\alpha) \geq T\}$;

(iii). the solution is $x_1^{\mathrm{LP}} = x_2^{\mathrm{LP}} = \dots = x_{k-1}^{\mathrm{LP}} = 1$, $x_k^{\mathrm{LP}} = (T - \sum_{i=1}^{k-1} f_i^\dagger(\alpha))/f_k^\dagger(\alpha)$, and $x_{k+1}^{\mathrm{LP}} = x_{k+2}^{\mathrm{LP}} = \dots = x_N^{\mathrm{LP}} = 0$.

A 2-factor solution to the binary KP (4) is obtained by performing the first two steps in the above procedure, and then picking the better of the two sets $\{1,\dots,k-1\}$ and $\{k\}$ (see [15, 12] for details). Let $F_{\mathrm{approx}} : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ be the value of the objective function in the α-parametrized knapsack problem under such a 2-factor solution.

If the optimal Lagrange multiplier α is known, then the aforementioned procedure can be used to determine a solution to (3) that is within a constant factor of the optimal. We now focus on the search for an efficient Lagrange multiplier α. We will show that an efficient solution can be computed by picking the maximizer of F_{LP} as the Lagrange multiplier. The maximizer of a continuous univariate function can be efficiently searched, but unfortunately, F_{LP} may admit several points of discontinuity. If the set of points of discontinuity is known, then the maximizer over each continuous piece can be searched efficiently. Therefore, we now determine the set of points of discontinuity of the function F_{LP}.

Lemma 3 (***Discontinuity of*** F_{LP})**.** *The maximal set of points of discontinuity of the function* F_{LP} *is* $\{\alpha_1,\dots,\alpha_N\}$.

Proof. For each $\alpha \in [0, \alpha_{\max}]$, the α-parametrized fractional KP is a linear program, and the solution lies at one of the vertex of the feasible simplex. Note that if $f_\ell^\dagger(\alpha)$ is a continuous function for each $\ell \in \{1,\dots,N\}$, then the vertices of the feasible simplex are continuous functions of α. Further, the objective function is also continuous if $f_\ell^\dagger(\alpha)$ is a continuous function for each $\ell \in \{1,\dots,N\}$. Therefore, F_{LP} may be discontinuous only if $f_\ell^\dagger(\alpha)$ is discontinuous for some ℓ, i.e., only if $\alpha \in \{\alpha_1,\dots,\alpha_N\}$. □

In summary, we will show that if each sigmoid function is consistent, then the allocation to each sigmoid function can be written in terms of the Lagrange multiplier α, and the KP with sigmoid utilities (3) reduces to the α-parametrized KP (4). Further, an efficient Lagrange multiplier α_{LP}^* can be searched in the interval $(0, \alpha_{\max}]$, and the α_{LP}^*-parametrized KP can be solved using standard approximation algorithms to determine a solution within a constant factor of the optimal. The search of an efficient Lagrange multiplier is a univariate continuous optimization problem and a typical optimization algorithm will converge only asymptotically, but it will converge to an arbitrarily small neighborhood of the efficient Lagrange multiplier in a finite number of iterations. Thus, a factor of optimality within an ϵ neighborhood of the desired factor of optimality, for any $\epsilon > 0$, can be achieved in a finite number of iterations.

In Algorithm 1, we utilize these ideas to obtain a solution within $(2+\epsilon)$-factor of the optimal solution for the KP with sigmoid utilities. The algorithm comprises four critical steps: (i) it searches for the Lagrange multiplier α_{LP}^* that maximizes F_{LP}; (ii) it determines a constant-factor solution to the α_{LP}^*-parametrized KP; (iii) it then compares $F_{\mathrm{approx}}(\alpha_{\mathrm{LP}}^*)$ with the values of the objective function corresponding to the allocations of the form $T\boldsymbol{e}_j$, $j \in \{1,\dots,N\}$, and picks the best among these policies; and (iv) it involves a performance-improvement heuristic in which the unemployed resource is allocated to the most beneficial item.

Note that step (iii) takes care of inconsistent sigmoid utilities. In particular, we will show that the allocation to an item with an inconsistent sigmoid utility is either zero or T, and thus, if a non-zero resource is allocated to an item with an inconsistent sigmoid utility, then every other item is allocated zero resource.

We now establish the performance of Algorithm 1. We define an ϵ-approximate maximizer of a function as a point in the domain of the function at which the function attains a value within ϵ of its maximum value. We now analyze Algorithm 1. We note that if the sigmoid utilities are non-smooth, then the standard KKT conditions in the following analysis are replaced with the KKT conditions for non-smooth optimization problems [29].

Theorem 4 (***KP with sigmoid utilities***)**.** *The following statements hold for the KP with sigmoid utilities* (3) *and the solution obtained via Algorithm 1:*

(i). *the solution is within a factor of optimality* $(2+\epsilon)$*, for any* $\epsilon > 0$*;*

Algorithm 1: *KP with Sigmoid Utilities: Approximation Algorithm*

```
Input  : $f_\ell, \ell \in \{1, \dots, N\}, T \in \mathbb{R}_{>0}$ ;
Output : optimal allocations $\boldsymbol{t}^* \in \mathbb{R}^N_{\geq 0}$;
   % search for optimal Lagrange multiplier
1  $\alpha^*_{\text{LP}} \leftarrow \operatorname{argmax}\{F_{\text{LP}}(\alpha) \mid \alpha \in [0, \alpha_{\max}]\}$;
2  determine the 2-factor solution $\boldsymbol{x}^*$ of $\alpha^*_{\text{LP}}$-parametrized knapsack problem ;
   % determine best inconsistent sigmoid function
3  find $\ell^* \leftarrow \operatorname{argmax}\{f_\ell(T) \mid \ell \in \mathcal{I}\}$;
   % pick the best among consistent and inconsistent tasks
4  if $f_{\ell^*}(T) > F_{\text{approx}}(\alpha^*_{\text{LP}})$ then
       $\boldsymbol{t}^* = T\boldsymbol{e}_{\ell^*}$ ;
5  else
       $t^\dagger_\ell \leftarrow x^*_\ell f^\dagger_\ell(\alpha^*_{\text{LP}}), \forall \ell \in \{1, \dots, N\}$;
       % heuristic to improve performance
       % pick the best sigmoid function to allocate the remaining resource
6      $\bar{\ell} \leftarrow \operatorname{argmax}\ \{f_\ell(t^\dagger_\ell + T - \sum_{j=1}^N t^\dagger_j) - f_\ell(t^\dagger_\ell) \mid \ell \in \{1, \dots, N\}\}$;
       $t^*_\ell \leftarrow \begin{cases} t^\dagger_\ell, & \text{if } \ell \in \{1, \dots, N\} \setminus \bar{\ell} \\ t^\dagger_\ell + T - \sum_{j=1}^N t^\dagger_j, & \text{if } \ell = \bar{\ell}; \end{cases}$
```

(ii). *if an ϵ-approximate maximizer over each continuous piece of F_{LP} can be searched using a constant number of function evaluations, then Algorithm 1 runs in $O(N^2)$ time.*

Proof. See Appendix. □

Corollary 5 (***Identical sigmoid functions***)**.** *If the sigmoid utilities in the KP with sigmoid utilities* (3) *are identical and equal to f, then an optimal solution $\boldsymbol{t}^*$ is an N-tuple with m^* entries equal to T/m^* and all other entries zero, where*

$$m^* = \operatorname*{argmax}_{m \in \{1, \dots, N\}} \quad m\, f(T/m). \tag{6}$$

Proof. It follows from Algorithm 1 that for identical sigmoid utilities the optimal non-zero resource allocated is the same for each item. The number of items with the optimal non-zero resource is determined by equation (6), and the statement follows. □

Discussion 1 (***Search of the optimal Lagrange multiplier***)**.** The approximate solution to the KP with sigmoid utilities in Algorithm 1 involves the search for α^*_{LP}, the maximizer of function F_{LP}. It follows from Lemma 3 that this search corresponds to the global maximization of N univariate continuous functions. The global maximum over each continuous piece can be determined using the P-algorithm [30, 31]. If stronger properties of F_{LP} can be established for a given instance of the KP with sigmoid utilities, then better algorithms can be utilized, e.g., (i) if each continuous piece of F_{LP} is differentiable, then the modified P-algorithm [32] can be used for global optimization; (ii) if each continuous piece of F_{LP} is Lipschitz, then one of the algorithms in [33] can be used for global optimization. □

Example 2. Given sigmoid functions $f_\ell(t) = w_\ell/(1 + \exp(-a_\ell t + b_\ell)), \ell \in \{1, \dots, 10\}$ with parameters and associated weights

$$\begin{aligned} \boldsymbol{a} &= (a_1, \dots, a_{10}) = (1, 2, 1, 3, 2, 4, 1, 5, 3, 6), \\ \boldsymbol{b} &= (b_1, \dots, b_{10}) = (5, 10, 3, 9, 8, 16, 6, 30, 6, 12), \\ \text{and } \boldsymbol{w} &= (w_1, \dots, w_{10}) = (2, 5, 7, 4, 9, 3, 5, 10, 13, 6). \end{aligned}$$

Let the total available resource be $T = 15$ units. The optimal solution and the approximate solution without the heuristic in step 6 of Algorithm 1 are shown in Figure 4. The approximate solution with the heuristic in step 6 of Algorithm 1 is the same as the optimal solution. The value functions F, F_{approx}, and F_{LP} are shown in Figure 5. □

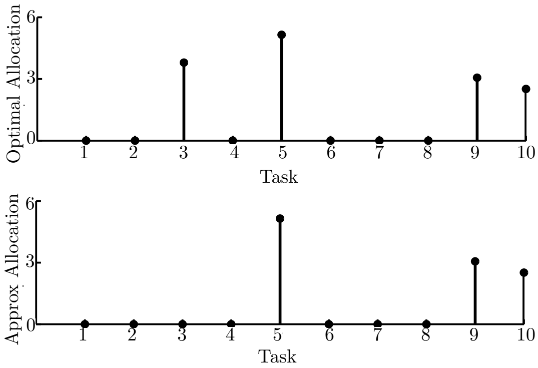

Figure 4: Optimal allocations and the approximate optimal allocations without the performance-improvement heuristic.

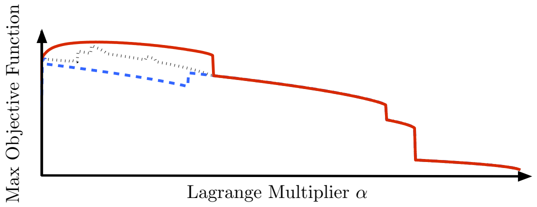

Figure 5: Exact and approximate maximum value of the objective function. The functions $F_{\text{LP}}, F, F_{\text{approx}}$ are shown by solid brown line, black dotted line, and blue dashed line, respectively. The points of discontinuity of the function F_{LP} are contained in the set $\{\alpha_1, \dots, \alpha_N\}$.

Remark 3 (***Multiple-choice KP with sigmoid utilities***)**.** Consider m disjoint classes $\{N_1, \dots, N_m\}$ of items and a single knapsack. The multiple-choice KP is to select one item each from every class such that the total utility of the selected items is maximized for a given total available resource. Let the total available resource be $T \in \mathbb{R}_{>0}$, and let the utility of allocating a resource $t \in \mathbb{R}_{\geq 0}$ to item i in class N_j be a sigmoid function $f_{ij} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The multiple-choice KP with sigmoid utilities is posed as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} \sum_{j \in N_i} f_{ij}(t_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{m} \sum_{j \in N_i} t_{ij} x_{ij} \leq T \\ & \sum_{j \in N_i} x_{ij} = 1, \ i \in \{1, \dots, m\} \\ & x_{ij} \in \{0, 1\}, \ i \in \{1, \dots, m\}, j \in N_i. \end{aligned} \tag{7}$$

Given a set of classes of tasks, the multiple-choice KP with sigmoid utilities models a situation where a human operator has to process one task each from every class within time T. The performance of the operator on task i from class N_j is given by the sigmoid function f_{ij}. Different tasks in a given class may be, e.g., observations collected from different sensors in a given region. The methodology developed in this section extends to the multiple-choice KP with sigmoid utilities (7). In particular, problem (7) can be reduced to an α-parameterized multiple-choice knapsack problem, and the LP relaxation based 2-factor approximation algorithm for the binary multiple choice knapsack problem [15] can be utilized to determine a 2-factor algorithm for problem (7). □

Remark 4 (***Allocation in queues with sigmoid utilities***). The KP with sigmoid utilities (3) also models the resource allocation problem in queues with sigmoid server performance functions. In particular, consider a single server queue with a general arrival process and a deterministic service process. Let the tasks arrive according to some process with a mean arrival rate λ. Let the tasks be indexed by the set $\{1, \dots, N\}$, and let each arriving task be sampled from a stationary probability vector $\{p_1, \dots, p_N\}$, i.e., at any time the next task arriving to the queue is indexed ℓ with probability p_ℓ. Let the performance of the server on a task with index ℓ be a sigmoid function f_ℓ of the service time. A stationary policy for such a queue always allocates a fixed duration $t_\ell \in \mathbb{R}_{\geq 0}$ to a task with index ℓ. An optimal stationary policy is a stationary policy that maximizes the expected performance of the server while keeping the queue stable. The stability constraint on the queue implies that the average allocation to each task should be smaller than $1/\lambda$. Accordingly, the optimal stationary policy is determined by:

$$\begin{aligned} \underset{t \geq 0}{\text{maximize}} \quad & \sum_{\ell=1}^{N} p_\ell f_\ell(t_\ell) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} p_\ell t_\ell \leq \frac{1}{\lambda}, \end{aligned}$$

which is a KP with sigmoid utilities. □

4. Generalized Assignment Problem with Sigmoid Utilities

In this section, we consider the GAP with sigmoid utilities. We first define the problem and then develop an approximation algorithm for it.

4.1. GAP with Sigmoid Utilities: Problem Description

Consider M bins (knapsacks) and N items. Let T_j be the total available resource at bin $j \in \{1, \dots, M\}$. Let the utility of item $i \in \{1, \dots, N\}$ when assigned to bin j be a sigmoid function $f_{ij} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ of the allocated resource t_{ij}. The GAP with sigmoid utilities determines the optimal assignment of the items to the bins such that the total utility of the bins is maximized. Note that unlike the assignment problem, the generalized assignment problem does not require every item to be allocated to some bin. Formally, the GAP with sigmoid utilities is posed as:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{M} \sum_{i=1}^{N} f_{ij}(t_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{N} t_{ij} x_{ij} \leq T_j, \ j \in \{1, \dots, M\} \\ & \sum_{j=1}^{M} x_{ij} \leq 1, \ i \in \{1, \dots, N\} \\ & x_{ij} \in \{0, 1\}, \ i \in \{1, \dots, N\}, j \in \{1, \dots, M\}. \end{aligned} \tag{8}$$

The GAP with sigmoid utilities models a situation where M human operators have to independently serve N tasks. The performance of operator j on task i is given by the sigmoid function f_{ij}, and she works for a total duration T_j. The solution to the GAP determines optimal assignments of the tasks to the operators and the associated optimal duration allocations. We now state the following result about the hardness of the GAP with sigmoid utilities:

Proposition 6 (***Hardness of GAP with sigmoid utilities***). *The GAP with sigmoid utilities is NP-hard, unless P = NP.*

Proof. The statement follows from the fact that the KP with sigmoid utilities is a special case of the GAP with sigmoid utilities, and is NP-hard according to Proposition 2. □

4.2. GAP with Sigmoid Utilities: Approximation Algorithm

We now propose an approximation algorithm for the GAP with sigmoid utilities. This algorithm is an adaptation of the 3-factor algorithm [34] for the binary GAP and is presented in Algorithm 2. We first introduce some notation. Let F be the matrix of sigmoid functions $f_{ij}, i \in \{1, \dots, N\}, j \in \{1, \dots, M\}$. Let $F_{*\ell}$ denote the ℓ-th column of the matrix F. For a given matrix E, let us denote $E_{*k:m}, k \leq m$ as the sub-matrix of E comprising of all the columns ranging from the k-th column to the m-th column. For a given set of allocations $t_{ij}, i \in \{1, \dots, N\}, j \in \{1, \dots, M\}$ and a set $\bar{A} \subseteq \{1, \dots, N\}$, $t_{\bar{A}j}$ represents the vector with entries $t_{ij}, i \in \bar{A}$. Similarly, for a given set $I_{\text{unproc}} \subseteq \{1, \dots, N\}$, $F_{I_{\text{unproc}}j}$ represents the vector with entries $F_{ij}, i \in I_{\text{unproc}}$. Let $\texttt{KP}(\cdot, \cdot)$ be the function which takes a set of sigmoid utilities and the total available resource as inputs and yields allocations according to Algorithm 1.

Algorithm 2 calls a recursive function $\texttt{GAP}(\cdot, \cdot)$ with the input $(1, F)$ to compute an approximate solution to the GAP with sigmoid utilities. The output of Algorithm 2 comprises a set A

describing assignments of the items to the bins and a matrix $\boldsymbol{t}$ describing the associated duration allocations.

The function $\texttt{GAP}(\cdot, \cdot)$ takes an index $\ell \in \{1, \dots, M\}$ and the matrix of sigmoid utilities $f_{ij}^{\ell}, i \in \{1, \dots, N\}, j \in \{\ell, \dots, M\}$ as the input and yields assignments of the items to the bin set $\{\ell, \dots, M\}$ and the associated duration allocations. The function $\texttt{GAP}(\ell, F^{(\ell)})$ first determines a temporary set of assignments and the associated duration allocations for the ℓ-th bin using Algorithm 1 with the sigmoid utilities in the first column of $F^{(\ell)}$ and the total available resource at the ℓ-th bin.

The function GAP then decomposes the matrix $F^{(\ell)}$ into two matrices E^1 and E^2 such that $F^{(\ell)} = E^1 + E^2$. The matrix E^1 is constructed by (i) picking its first column as the first column of $F^{(\ell)}$, (ii) picking the remaining entries of the rows associated with the items temporarily assigned to the ℓ-th bin as the value of the sigmoid function in the first column computed at the associated temporary allocation, and (iii) picking all other entries as zero. The matrix E^2 is chosen as $E^2 = F^{(\ell)} - E^1$. The key idea behind this decomposition is that the matrix E^2 has all the entries in the first column equal to zero, and thus, effectively contains only $M - \ell$ columns of sigmoid utilities.

The function GAP then removes the first column of E^2 , assigns the resulting matrix to $F^{(\ell+1)}$, and calls itself with the input $(\ell + 1, F^{(\ell+1)})$. The recursion stops at $\ell + 1 = M$, in which case $F^{(\ell+1)}$ is a column vector, and the assignments with the associated allocations are obtained using Algorithm 1.

Algorithm 2 also involves a performance-improving heuristic. According to this heuristic, if the total available resource at a bin is not completely utilized and there are tasks that are not assigned to any bin, then a KP with sigmoid utilities is solved using the remaining amount of the resource and unassigned tasks. Likewise, if the total available resource at a bin is not completely utilized and each task has been assigned to some bin, then the remaining resource is allocated to the most beneficial task in that bin.

We now establish performance bounds for the proposed algorithm:

Theorem 7 (***GAP with sigmoid utilities***)**.** *The following statements hold for the GAP with sigmoid utilities* (8) *and the solution obtained via Algorithm 2:*

(i). *the solution is within a factor* $(3 + \epsilon)$ *of the optimal, for any* $\epsilon > 0$*; and*

(ii). *Algorithm 2 runs in* $O(N^2 M)$ *time, provided the solution to the KP with sigmoid utilities can be computed in* $O(N^2)$ *time.*

Proof. The proof is an adaptation of the inductive argument used in [34] to establish the performance of a similar algorithm for the binary GAP. We note that for a single bin, the GAP reduces to the knapsack problem and Algorithm 1 provides a solution within $(2 + \epsilon)$-factor of the optimal. Consequently, Algorithm 2 provides a solution within $(2 + \epsilon)$-factor of the optimal, and hence, within $(3 + \epsilon)$-factor of the optimal.

Algorithm 2: *GAP with Sigmoid Utilities: 3-factor Approximation*

```
Input  : $f_{ij}, T_j, i \in \{1, \dots, N\}, j \in \{1, \dots, M\}$ ;
Output : assignment set $A = \{A_1, \dots, A_M\}$ and allocations $\boldsymbol{t} \in \mathbb{R}_{\geq 0}^{N \times M}$;

% Initialize
$F^{(1)} \leftarrow F$;
% Call function GAP
allocations $[A, \boldsymbol{t}] \leftarrow \texttt{GAP}(1, F^{(1)})$;
% heuristic to improve performance
% assign unassigned tasks to unsaturated bins
$I_{\text{unproc}} \leftarrow \{1, \dots, N\} \setminus \cup_{k=1}^{M} A_k$;
foreach $j \in \{1, \dots, M\}$ do
    if $\sum_{i \in A_j} t_{ij} < T_j$ and $|I_{\text{unproc}}| > 0$ then
        % solve KP with unprocessed tasks
        $[\bar{A}, \bar{\boldsymbol{t}}] \leftarrow \texttt{KP}(F_{I_{\text{unproc}} j}, T_j - \sum_{i \in A_j} t_{ij})$;
        $A_j \leftarrow A_j \cup \bar{A}$;    $t_{\bar{A} j} \leftarrow \bar{t}$;
    else if $\sum_{i \in A_j} t_{ij} < T_j$ and $|I_{\text{unproc}}| = 0$ then
        % allocate remaining resource to the most rewarding task
        $\upsilon \leftarrow \operatorname{argmax}\{f_{ij}(t_{ij} + T_j - \sum_{i \in A_j} t_{ij}) \mid i \in A_j\}$;
        $t_{\upsilon j} \leftarrow t_{\upsilon j} + T_j - \sum_{i \in A_j} t_{ij}$;
    $I_{\text{unproc}} \leftarrow \{1, \dots, N\} \setminus (A_1 \cup \dots \cup A_m)$ ;
-----------------------------------------------------------------
% Function definition
function $[A^{(\ell)}, \boldsymbol{t}^{(\ell)}] \leftarrow \texttt{GAP}(\ell, F^{(\ell)})$
% Determine temporary the allocations for bin $\ell$ using Algorithm 1
$[\bar{A}, \bar{\boldsymbol{t}}] \leftarrow \texttt{KP}(F_{*1}^{(\ell)}, T_\ell)$;
foreach $i \in \{1, \dots, N\}$ and $j \in \{1, \dots, M - \ell + 1\}$ do
    $E_{ij}^1(t) \leftarrow \begin{cases} F_{i1}^{(\ell)}(\bar{t}_i), & \text{if } i \in \bar{A} \text{ and } j \neq 1, \\ F_{i1}^{(\ell)}(t), & \text{if } j = 1, \\ 0, & \text{otherwise;} \end{cases}$
$E^2(t) \leftarrow F^{(\ell)}(t) - E^1(t)$;
if $\ell < M$ then
    % remove first column from $E^2$ and assign it to $F^{(\ell+1)}$
    $F^{(\ell+1)} \leftarrow E_{*2:M-\ell+1}^2$;
    $[A^{(\ell+1)}, \boldsymbol{t}^{(\ell+1)}] \leftarrow \texttt{GAP}(\ell + 1, F^{(\ell+1)})$;
    $A_\ell \leftarrow \bar{A} \setminus \cup_{k=\ell+1}^{M} A_k$;
    $A^{(\ell)} \leftarrow A_\ell \cup A^{(\ell+1)}$;
    foreach $i \in \bar{A} \cap \cup_{k=\ell+1}^{M} A_k$ do
        $\bar{t}_i \leftarrow 0$;
    $\boldsymbol{t}^{(\ell)} \leftarrow [\bar{\boldsymbol{t}} \quad \boldsymbol{t}^{(\ell+1)}]$;
else
    $A_\ell \leftarrow \bar{A}$ and $\boldsymbol{t}^{(\ell)} \leftarrow \bar{\boldsymbol{t}}$;
```

Assume by the induction hypothesis that Algorithm 2 provides a solution within $(3 + \epsilon)$-factor of the optimal for L bins. We now consider the case with $(L+1)$ bins. The performance matrix F has two components, namely, E^1 and E^2. We note that first column of E^2 has each entry equal to zero, and thus, E^2 corresponds to a GAP with L bins. By the induction hypothesis, Algorithm 2 provides a solution within $(3+\epsilon)$-factor of the optimal with respect to performance matrix E^2. We further note that the first column of E^1 is identical to the first column of F and Algorithm 1 provides a solution within $(2 + \epsilon)$-factor of the optimal with respect to this column (bin). Moreover, the best possible allocation with respect to other entries can contribute to the objective function an amount at most equal to $\sum_{i=1}^{N} f_{i1}(t_{i1}^*)$. Consequently, the solution obtained from Algorithm 2 is within $(3+\epsilon)$-factor of the optimal with respect to performance matrix E^1. Since the solution is within $(3+\epsilon)$-factor of the optimal with respect to both E^1 and E^2, it follows that the solution is within $(3+\epsilon)$-factor of the optimal with respect to $E^1 + E^2$ (see Theorem 2.1 in [34]). The performance improvement heuristic further improves the value of the objective function and improves the factor of optimality. Consequently, the established factor of optimality still holds. This establishes the first statement.

The second statement follows immediately from the observation that Algorithm 2 solves $2M$ instances of knapsack problem with sigmoid utilities using Algorithm 1. □

Example 3. Consider the GAP with $M = 4$ bins and $N = 10$ items. Let the sigmoid utility associated with bin i and item j be $f_{ij}(t) = 1/(1+\exp(-t+b_{ij}))$, where the matrix of parameters b_{ij} is

$$\boldsymbol{b} = \begin{bmatrix} 1 & 7 & 2 & 3 & 8 & 7 & 5 & 1 & 3 & 6 \\ 7 & 9 & 8 & 8 & 6 & 1 & 7 & 4 & 5 & 4 \\ 6 & 10 & 1 & 2 & 3 & 1 & 9 & 7 & 9 & 5 \\ 9 & 2 & 4 & 8 & 1 & 2 & 5 & 8 & 6 & 8 \end{bmatrix}.$$

Let the vector of the total resource available at each bin be $\boldsymbol{T} = [5\ 10\ 15\ 20]$. The resource allocations to different items obtained using Algorithm 2 are shown in Figure 6. The assignment sets of items to bins are $A_1 = \{8\}, A_2 = \{10\}, A_3 = \{1,3,4,5\}$, and $A_4 = \{2,6,7,9\}$. □

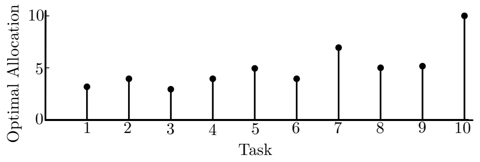

Figure 6: Allocations for the GAP obtained using Algorithm 2.

5. Bin-packing Problem with Sigmoid Utilities

In this section, we consider the BPP with sigmoid utilities. We first define the problem and then develop an approximation algorithm for it.

5.1. BPP with Sigmoid Utilities: Problem Description

Consider a set of N items with sigmoid utilities $f_\ell, \ell \in \{1,\dots,N\}$, and an unlimited number of bins, each with a resource $T \in \mathbb{R}_{>0}$. The BPP with sigmoid utilities determines the minimum number of bins $K \in \mathbb{N}$ and assignments of of items to bins $\Upsilon : \{1,\dots,N\} \to \{1,\dots,K\}$ such that the KP with sigmoid utilities associated with each bin and items assigned to it allocates a non-zero resource to each item in the bin. Formally, let $\mathcal{A}_i$ be the set of items assigned to bin $i \in \{1,\dots,K\}$, that is, $\mathcal{A}_i = \{j \in \{1,\dots,N\} \mid \Upsilon(j) = i\}$. Then, the BPP with sigmoid utilities finds the minimum K and sets $\mathcal{A}_i, i \in \{1,\dots,K\}$ such that the optimal solution to the following KP with sigmoid utilities, for each $i \in \{1,\dots,K\}$, allocates a non-zero resource to each item $\ell \in \mathcal{A}_i$:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell \in \mathcal{A}_i} f_\ell(t_\ell) \\ \text{subject to} \quad & \sum_{\ell \in \mathcal{A}_i} t_\ell \le T. \end{aligned} \tag{9}$$

The BPP with sigmoid utilities determines the minimum number of identical operators, each working for a total duration T, required to optimally serve each of the N tasks characterized by sigmoid functions $f_\ell, \ell \in \{1,\dots,N\}$.

We will establish that the standard BPP is a special case of the BPP with sigmoid utilities, and consequently, the BPP with sigmoid utilities is NP-hard. To this end, we need to determine an amount of the resource T such that each item in a given set $\mathcal{A}_i$ is allocated a non-zero resource by the solution to (9) obtained using Algorithm 1. We denote the critical penalty rate for the sigmoid function f_ℓ by $\psi_\ell, \ell \in \{1,\dots,N\}$, and let $\psi_{\min} = \min\{\psi_\ell \mid \ell \in \{1,\dots,N\}\}$.

Lemma 8 (***Non-zero allocations***)**.** *A solution to the optimization problem* (9) *allocates a non-zero resource to each sigmoid function f_ℓ, $\ell \in \mathcal{A}_i, i \in \{1,\dots,K\}$, if*

$$T \ge \sum_{\ell \in \mathcal{A}_i} f_\ell^\dagger(\psi_{\min}).$$

Proof. It suffices to prove that if $T = \sum_{\ell \in \mathcal{A}_i} f_\ell^\dagger(\psi_{\min})$, then $\psi_{\min}$ is the optimal Lagrange multiplier α_{LP}^* in Algorithm 1. Note that if a non-zero resource is allocated to each task, then the solution obtained from Algorithm 1 is the optimal solution to (3). Since, $t_\ell^* = f_\ell^\dagger(\psi_{\min}), \ell \in \mathcal{A}_i$ are feasible non-zero allocations, $\psi_{\min}$ is a Lagrange multiplier. We now prove that $\psi_{\min}$ is the optimal Lagrange multiplier. Let $\mathcal{A}_i = \{1,\dots,a_i\}$. By contradiction, assume that $\boldsymbol{t}^*$ is not the globally optimal allocation. Without loss of generality, we assume that the global optimal policy allocates zero resource to sigmoid function f_{a_i}, and let $\bar{\boldsymbol{t}}$ be the globally optimal allocation. We observe that

$$\begin{aligned} &\sum_{\ell=1}^{a_i-1} f_\ell(\bar{t}_\ell) + f_{a_i}(0) \\ &\quad \le \sum_{\ell=1}^{a_i-1} f_\ell(\bar{t}_\ell) + f_{a_i}(t_{a_i}^*) - \psi_{\min} t_{a_i}^* && (10) \\ &\quad \le \sum_{\ell=1}^{a_i} f_\ell(t_\ell^*) + \sum_{\ell=1}^{a_i-1} f_\ell'(t_\ell^*)(\bar{t}_\ell - t_\ell^*) - \psi_{\min} t_{a_i}^* && (11) \\ &\quad = \sum_{\ell=1}^{a_i} f_\ell(t_\ell^*) + \sum_{\ell=1}^{a_i} \psi_{\min}(\bar{t}_\ell - t_\ell^*) = \sum_{\ell=1}^{a_i} f_\ell(t_\ell^*) \end{aligned}$$

where inequalities (10) and (11) follow from the definition of the critical penalty rate and the concavity to the sigmoid function at t_ℓ^*, respectively. This contradicts our assumption. Hence, $\boldsymbol{t}^*$ is the global optimal allocation and this completes the proof. □

We now state the following result about the hardness of the BPP with sigmoid utilities:

Proposition 9 (***Hardness of the BPP with sigmoid utilities*****).** *The BPP with sigmoid utilities is NP-hard, unless P = NP.*

Proof. Consider an instance of the standard BPP with items of size $a_i \leq T, i \in \{1, \dots, N\}$ and bins of size T. It is well known [12] that the BPP is NP-hard. Without loss of generality, we can pick N sigmoid functions $f_i, i \in \{1, \dots, N\}$ such that $f_i^\dagger(\psi_{\min}) = a_i$, for each $i \in \{1, \dots, N\}$ and some $\psi_{\min} \in \mathbb{R}_{>0}$. It follows from Lemma 8 that such an instance of the BPP with sigmoid utilities is in a one-to-one correspondence with the aforementioned standard BPP. This establishes the statement. □

5.2. BPP with Sigmoid Utilities: Approximation Algorithm

We now develop an approximation algorithm for the BPP with sigmoid utilities. The proposed algorithm is similar to the standard next-fit algorithm [12] for the binary BPP. The algorithm iteratively performs three critical steps: (i) it adds an item to the current bin; (ii) if after the addition of the item, the optimal policy for the associated KP with sigmoid utilities allocates a non-zero resource to each item in the bin, then it assigns the item to the current bin; (iii) otherwise, it opens a new bin and allocates the item to the new bin. This approximation algorithm is presented in Algorithm 3. We now present a formal analysis of this algorithm. We introduce following notations. Let K^* be the number of bins used by the optimal solution to the bin-packing problem with sigmoid utilities, and let $K_{\text{next-fit}}$ be the number of bins used by the solution obtained through Algorithm 3.

Algorithm 3: *BPP with Sigmoid Utilities: Approx. Algorithm*

```
Input    : $f_\ell, \ell \in \{1, \dots, N\}, T \in \mathbb{R}_{>0}$ ;
Output   : number of required bins $K \in \mathbb{N}$ and assignments $\Upsilon$;

$K \leftarrow 1; \mathcal{A}_K \leftarrow \{\}$;
foreach $\ell \in \{1, \dots, N\}$ do
    $\mathcal{A}_K \leftarrow \mathcal{A}_K \cup \{\ell\}$ ;
    solve problem (9) for $i = K$, and find $\boldsymbol{t}^*$ ;
    % if optimal policy drops a task, open a new bin
    if $t_j^* = 0$, for some $j \in \mathcal{A}_K$ then
        $K \leftarrow K + 1; \mathcal{A}_K \leftarrow \{\ell\}$;
    $\Upsilon(\ell) \leftarrow K$;
```

Theorem 10 (***BPP with sigmoid utilities*****).** *The following statements hold for the BPP with sigmoid utilities* (9)*, and its solution obtained via Algorithm 3:*

(i). *the optimal solution satisfies the following bounds*

$$K_{\text{next-fit}} \geq K^* \geq \frac{1}{T} \sum_{\ell=1}^{N} \min\{T, t_\ell^{\text{inf}}\}.$$

(ii). *the solution obtained through Algorithm 3 satisfies*

$$K_{\text{next-fit}} \leq \frac{1}{T}\Big(2 \sum_{\ell=1}^{N} \min\{T, f_\ell^\dagger(\psi_{\min})\} - 1\Big).$$

(iii). *Algorithm 3 provides a solution to the BPP with sigmoid utilities within a factor of optimality*

$$\frac{\max\{2 \min\{T, f_\ell^\dagger(\psi_{\min})\} \mid \ell \in \{1, \dots, N\}\}}{\max\{\min\{T, t_\ell^{\text{inf}}\} \mid \ell \in \{1, \dots, N\}\}};$$

(iv). *Algorithm 3 runs in $O(N^3)$ time, provided the solution to the KP with sigmoid utilities can be computed in $O(N^2)$ time.*

Proof. It follows from Algorithm 1 that if $t_\ell^{\text{inf}} < T$, then the optimal non-zero allocation to the sigmoid function f_ℓ is greater than t_ℓ^{inf}. Otherwise, the optimal non-zero allocation is equal to T. Therefore, if each sigmoid function gets a non-zero allocation under the optimal policy, then at least $\sum_{\ell=1}^{N} \min\{T, t_\ell^{\text{inf}}\}$ resource is required, and the lower bound on the optimal K^* follows.

It follows from Lemma 8 that if $t_\ell = f_\ell^\dagger(\psi_{\min})$ amount of the resource is available for task ℓ, then a non-zero resource is allocated to it. Therefore, the solution of the bin-packing problem with bin size T and items of size $\{\min\{T, f_\ell^\dagger(\psi_{\min})\} \mid \ell \in \{1, \dots, N\}\}$ provides an upper bound to the solution of the BPP with sigmoid utilities. The upper bound to the solution of this bin-packing problem obtained through the standard next-fit algorithm is $(2 \sum_{\ell=1}^{N} \min\{T, f_\ell^\dagger(\psi_{\min})\} - 1)/T$, and this completes the proof of the second statement.

The third statement follows immediately from the first two statements, and the last statement follows immediately from the fact that Algorithm 1 is utilized at each iteration of Algorithm 3. □

Example 4. For the same set of sigmoid functions as in Example 2 and $T = 20$ units, the solution to the BPP with sigmoid utilities obtained through Algorithm 3 requires $K_{\text{next-fit}} = 3$ bins, and the associated allocations to each task in these bins are shown in Figure 7. □

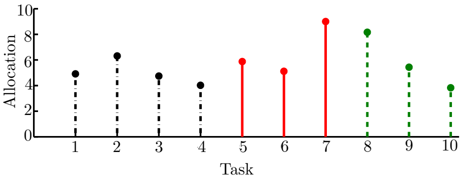

Figure 7: Allocations to items in each bin. The dot-dashed black lines represent items assigned to the first bin, the solid red lines represent items assigned to the second bin, and the dashed green line represent items assigned to the third bin.

6. Conclusions and Future Directions

We studied non-convex optimization problems involving sigmoid functions. We considered the maximization of a sigmoid function subject to a linear penalty and showed that the optimal allocation jumps down to zero at a critical penalty rate. This jump in the allocation imparts combinatorial effects to the constrained optimization problems involving sigmoid functions. We studied three such problems, namely, the KP with sigmoid utilities, the GAP with sigmoid utilities, and the BPP with sigmoid utilities. We merged approximation algorithms from discrete optimization with algorithms from continuous optimization to develop hybrid approximation algorithms for these problems.

There are many possible extensions of this work. A similar strategy for approximate optimization could be adopted for other problems involving sigmoid functions, e.g., the network utility maximization problem, where the utility of each source is a sigmoid function. Other extensions include problems involving general non-convex functions and optimization in general queues with sigmoid characteristics.

References

[1] V. Srivastava and F. Bullo. Hybrid combinatorial optimization: Sample problems and algorithms. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7212–7217, Orlando, FL, USA, December 2011.

[2] E. Guizzo. Obama commanding robot revolution announces major robotics initiative. IEEE Spectrum, June 2011.

[3] W. M. Bulkeley. Chicago's camera network is everywhere. The Wall Street Journal, November 17, 2009.

[4] C. Drew. Military taps social networking skills. The New York Times, June 7, 2010.

[5] T. Shanker and M. Richtel. In new military, data overload can be deadly. The New York Times, January 16, 2011.

[6] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen. The physics of optimal decision making: A formal analysis of performance in two-alternative forced choice tasks. *Psychological Review*, 113(4):700–765, 2006.

[7] R. W. Pew. The speed-accuracy operating characteristic. *Acta Psychologica*, 30:16–26, 1969.

[8] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, 3 edition, 2000.

[9] S. K. Hong and C. G. Drury. Sensitivity and validity of visual search models for multiple targets. *Theoretical Issues in Ergonomics Science*, 3(1):85–110, 2002.

[10] D. Vakratsas, F. M. Feinberg, F. M. Bass, and G. Kalyanaram. The shape of advertising response functions revisited: A model of dynamic probabilistic thresholds. *Marketing Science*, 23(1):109–119, 2004.

[11] M. H. Rothkopf. Bidding in simultaneous auctions with a constraint on exposure. *Operations Research*, 25(4):620–629, 1977.

[12] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithmics and Combinatorics*. Springer, 4 edition, 2007.

[13] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, 1990.

[14] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, 1988.

[15] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[16] K. M. Bretthauer and B. Shetty. The nonlinear knapsack problem–algorithms and applications. *European Journal of Operational Research*, 138(3):459–472, 2002.

[17] S. Kameshwaran and Y. Narahari. Nonconvex piecewise linear knapsack problems. *European Journal of Operational Research*, 192(1):56–68, 2009.

[18] J. J. Moré and S. A. Vavasis. On the solution of concave knapsack problems. *Mathematical Programming*, 49(1):397–411, 1990.

[19] G. J. Burke, J. Geunes, H. E. Romeijn, and A. Vakharia. Allocating procurement to capacitated suppliers with concave quantity discounts. *Operations Research Letters*, 36(1):103–109, 2008.

[20] W. Ginsberg. The multiplant firm with increasing returns to scale. *Journal of Economic Theory*, 9(3):283–292, 1974.

[21] J. R. Freeland and C. B. Weinberg. S-Shaped response functions: Implications for decision models. *Journal of the Operational Research Society*, 31(11):1001–1007, 1980.

[22] S. Ağrali and J. Geunes. Solving knapsack problems with S-curve return functions. *European Journal of Operational Research*, 193(2):605–615, 2009.

[23] V. Srivastava, R. Carli, F. Bullo, and C. Langbort. Task release control for decision making queues. In *American Control Conference*, pages 1855–1860, San Francisco, CA, USA, June 2011.

[24] V. Srivastava, R. Carli, C. Langbort, and F. Bullo. Attention allocation for decision making queues. *Automatica*, February 2012. to appear.

[25] L. F. Bertuccelli, N. W. M. Beckers, and M. L. Cummings. Developing operator models for UAV search scheduling. In *AIAA Conf. on Guidance, Navigation and Control*, Toronto, Canada, August 2010.

[26] A. G. Rao and M. R. Rao. Optimal budget allocation when response is S-shaped. *Operations Research Letters*, 2(5):225–230, 1983.

[27] D. N. Southern. Human-guided management of collaborating unmanned vehicles in degraded communication environments. Master's thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2010.

[28] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2):183–238, 1993.

[29] J. B. Hiriart-Urruty. On optimality conditions in nondifferentiable programming. *Mathematical Programming*, 14(1):73–86, 1978.

[30] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[31] J. Calvin. Convergence rate of the P-algorithm for optimization of continuous functions. In P. M. Pardalos, editor, *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*. Kluwer Academic, 1999.

[32] J. Calvin and A. Žilinskas. On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions. *Journal of Optimization Theory and Applications*, 102(3):479–495, 1999.

[33] P. Hansen, B. Jaumard, and S. H. Lu. Global optimization of univariate Lipschitz functions: I. Survey and properties. *Mathematical Programming*, 55(1):251–272, 1992.

[34] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4):162–166, 2006.

[35] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 2 edition, 1984.

Appendix

A-1. ***Proof of Theorem 4***

We apply the Karush-Kuhn-Tucker necessary conditions [35] for an optimal solution:
Linear dependence of gradients

$$\frac{\partial L}{\partial t_\ell^*}(\boldsymbol{t}^*, \alpha^*, \boldsymbol{\mu}^*) = f_\ell'(t_\ell^*) - \alpha^* + \mu_\ell^* = 0, \text{ for each } \ell \in \{1, \dots, N\}. \tag{A.1}$$

Feasibility of the solution

$$T - \mathbf{1}_N^{\mathrm{T}} \boldsymbol{t}^* \geq 0 \quad \text{and } \boldsymbol{t}^* \geq 0. \tag{A.2}$$

Complementarity conditions

$$\alpha^*(T - \mathbf{1}_N^{\mathrm{T}}\boldsymbol{t}^*) = 0. \tag{A.3}$$

$$\mu_\ell^* t_\ell^* = 0, \quad \text{for each } \ell \in \{1, \dots, N\}. \tag{A.4}$$

Non-negativity of the multipliers

$$\alpha^* \geq 0, \quad \boldsymbol{\mu}^* \geq 0. \tag{A.5}$$

Since f_ℓ is a non-decreasing function, for each $\ell \in \{1, \dots, N\}$, the resource constraint should be active, and thus, from complementarity condition (A.3) $\alpha^* > 0$. Further, from equation (A.4), if $t_\ell^* \neq 0$, then $\mu_\ell^* = 0$. Therefore, if a non-zero resource is allocated to the sigmoid function $f_\eta, \eta \in \{1, \dots, N\}$, then it follows from equation (A.1)

$$f_\eta'(t_\eta^*) = \alpha^*. \tag{A.6}$$

Assuming that each f_ℓ is consistent, i.e., $t_\ell^{\inf} \leq T$, for each $\ell \in \{1, \dots, N\}$, the second order condition [35] yields that a local maxima exists at $\boldsymbol{t}^*$ only if

$$f_\eta''(t_\eta^*) \leq 0 \iff t_\eta^* \geq t_\eta^{\inf}. \tag{A.7}$$

The equations (A.6) and (A.7) yield that the optimal non-zero allocation to the sigmoid function f_η is

$$t_\eta^* = f_\eta^\dagger(\alpha^*). \tag{A.8}$$

Given the optimal Lagrange multiplier α^*, the optimal non-zero allocation to the sigmoid function f_η is given by equation (A.8). Further, the optimal set of sigmoid functions with non-zero allocations is the solution to the α^*-parametrized KP (4). We now show that α^* is the maximizer of F. Since, at least one task is processed, $f_\ell'(t_\ell^*) = \alpha$, for some $\ell \in \{1, \dots, N\}$. Thus, $\alpha \in [0, \alpha_{\max}]$. By contradiction assume that $\bar{\alpha}$ is the maximizer of F, and $F(\bar{\alpha}) > F(\alpha^*)$. This means that the allocation corresponding to $\bar{\alpha}$ yields higher reward than the allocation corresponding to α^*. This contradicts equation (A.8).

If $t_\ell^{\inf} > T$, for some $\ell \in \{1, \dots, N\}$, then equation (A.7) does not hold for any $t_\ell \in [0, T]$. Since, f_ℓ is convex in the interval $[0, T]$, the optimal allocation is at the boundary, i.e., $t_\ell \in \{0, T\}$. Therefore, as exemplified in Figure 8, the optimal allocation is either $T\boldsymbol{e}_\ell$ or lies at the projection of the simplex on the hyperplane $t_\ell = 0$. The projection of the simplex on the hyperplane $t_\ell = 0$ is again a simplex and the argument holds recursively.

To establish the first statement we note that α_{LP}^* is the maximizer of F_{LP}, and the α-parametrized fractional KP is a relaxation of the α-parametrized KP, hence

$$F_{\mathrm{LP}}(\alpha_{\mathrm{LP}}^*) \geq F_{\mathrm{LP}}(\alpha^*) \geq F(\alpha^*). \tag{A.9}$$

We further note that α^* is the maximizer of F and F_{approx} is a suboptimal value of the objective function, hence

$$F(\alpha^*) \geq F(\alpha_{\mathrm{LP}}^*) \geq F_{\mathrm{approx}}(\alpha_{\mathrm{LP}}^*) \geq \frac{1}{2} F_{\mathrm{LP}}(\alpha_{\mathrm{LP}}^*), \tag{A.10}$$

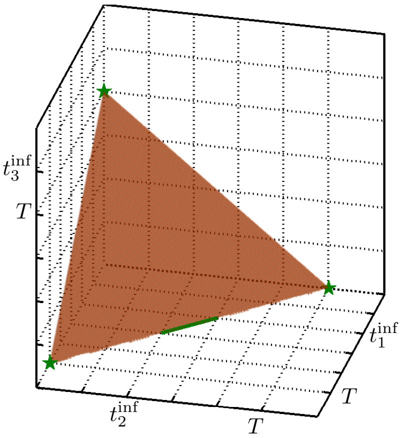

Figure 8: Possible locations of the maximum are shown in green stars and solid green line. The maximum possible allocation T is smaller than the inflection point of the third sigmoid function. For any allocation to the third sigmoid function, the corresponding entry in the Hessian matrix is positive, and the optimal allocation to the third sigmoid function is 0 or T. Optimal allocation to the first and the second sigmoid function may lie at the vertex of the simplex, or at a location where the Jacobian is zero and the Hessian matrix is negative definite.

where the last inequality follows from the construction of F_{approx} (see 2-factor policy for the binary knapsack problem in [12]). The value of the objective function at $\boldsymbol{t}^\dagger$ in Algorithm 1 is equal to $F_{\mathrm{approx}}(\alpha_{\mathrm{LP}}^*)$. The allocation $\boldsymbol{t}^\dagger$ may not saturate the entire resource T. Since, the sigmoid functions are non-decreasing with the allocated resource, entire resource must be utilized, and it is heuristically done in step 6 of Algorithm 1. This improves the value of the objective function and the factor of optimality remains at most 2. Finally, since a numerical method will only compute an ϵ-approximate maximizer of F_{LP} in finite time, the factor of optimality increases to $(2 + \epsilon)$.

To establish the last statement, we note that each evaluation of F_{LP} requires the solution of the α-parametrized fractional KP and has $O(N)$ computational complexity. According to Lemma 3, the maximum number of points of discontinuity of F_{LP} is $N + 1$. Therefore, if ϵ-approximate maximizer over each continuous piece of F_{LP} can be searched using a constant number of function evaluations, then $O(N)$ computations are needed over each continuous piece of F_{LP}. Consequently, the Algorithm 1 runs in $O(N^2)$ time. □