Through the Lens of Sequence Submodularity

Sara Bernardini
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey, UK, TW20 0EX
sara.bernardini@rhul.ac.uk

Fabio Fagnani
Department of Mathematical Sciences
Politecnico di Torino
Torino, Italia, 10129
fabio.fagnani@polito.it

Chiara Piacentini
Augmenta Inc.
Toronto, Canada, M5A 1E1
chiara@augmenta.ai

Abstract

Several real-world problems in engineering and applied science require the selection of *sequences* that maximize a given reward function. Optimizing over sequences as opposed to sets requires exploring an exponentially larger search space and can become prohibitive in most cases of practical interest. However, if the objective function is submodular (intuitively, it exhibits a diminishing return property), the optimization problem becomes more manageable. Recently, there has been increasing interest in *sequence submodularity* in connection with applications such as recommender systems and online ad allocation. However, mostly ad hoc models and solutions have emerged within these applicative contexts. In consequence, the field appears fragmented and lacks coherence. In this paper, we offer a unified view of sequence submodularity and provide a generalized greedy algorithm that enjoys strong theoretical guarantees. We show how our approach naturally captures several application domains, and our algorithm encompasses existing methods, improving over them.

Introduction

Many real-world applications in engineering and applied science have at their core the selection of *sequences* of objects that maximize a reward. In information gathering missions, for example, the objects are observations and the goal is to select a sequence of them that maximizes the information gain (Krause and Guestrin 2007; Piacentini, Bernardini, and Beck 2019). In a similar fashion, a movie recommender system aims to provide its users with sequences of items that maximize relevance (Ashkan et al. 2015; Tschiatschek, Singla, and Krause 2017). The crucial point in these applications is that the value of the sequence depends not only on the objects belonging to it, but also on their relative order. This is because the value of each object changes based on its position in the sequence.

If optimizing over sets is already a daunting task, optimizing over sequences quickly becomes intractable when the problem at hand grows. However, the identification of special properties in the objective function helps in making the task more approachable. *Submodularity*, in particular, has emerged as a powerful feature that can be leveraged to control complexity in the maximization of both set and sequence functions. Submodularity can be understood intuitively as a *diminishing return* condition. Consider again an information-gathering mission. It is easy to see that each new observation increases the information gain, but to a smaller extent than the previous observations, with gain vanishing at infinity.

In areas as variegated as optimization, machine learning, economics, medicine and sensor networks, there has been a vast amount of work on the maximization of submodular set functions (see Related Work). Only recently, the scientific community has started to pay closer attention to *sequence submodularity* prompted by applications such as online ad allocation (Alaei and Malekian 2010) and recommendations in online shopping (McAuley, Pandey, and Leskovec 2015), entertainment (Ashkan et al. 2015) and courses (Mitrovic et al. 2018). However, having arisen in specific applicative contexts, the proposed models as well as the corresponding algorithms lack generality and require making restrictive assumptions on the objective function to maintain efficiency.

In this paper, to remedy the current ad-hoc approach and lack of coherence in the field, we offer a unified view of sequence submodularity. By abstracting away specific applicative details, we show that the optimization problem that lies behind several applications can be captured by a particular type of *recursive submodular functional*. We study its structure and, based on its properties, we propose a *generalized greedy algorithm* that has theoretical guarantees as strong as its classical counterpart on set functions, but does not require unrealistic restrictive assumptions. Our generalized algorithm encompasses and improves the specific algorithms that have been developed for several practical applications. Another property that confers *flexibility* to our approach is that we can easily enforce constraints on the cardinality of the elements in the sequence (e.g. all elements must be distinct) in the domain description, which is particularly useful in applicative problems.

The paper is organized as follows. After discussing related work, we state the problem formally and introduce three, very different application domains, which demonstrate the expressiveness and generality of our approach. We then define and study submodularity for sequence functions showing how, in general, a simple generalization from sets to sequences of the classical greedy algorithm fails to achieve good performance. Subsequently, we propose and analyze a new greedy algorithm that is proven to achieve the same performance as the classical one for submodular set functions (Theorem 2). After presenting how this result can be used in our applicative examples (Theorem 3), we conclude the paper with explicit numerical simulations for one of them.

Related Work

Work on submodularity spreads across multiple fields, including optimization (Fujishige 2005; Nemhauser and Wolsey 1978), machine learning (Golovin and Krause 2011; Krause and Guestrin 2005), economics (Dughmi, Roughgarden, and Sundararajan 2012; Lehmann, Lehmann, and Nisan 2006), medicine (Hoi et al. 2006) and sensor networks (Zhang et al. 2016; Krause, Singh, and Guestrin 2008). This body of work focuses on *set functions* and, as most of the problems considered are NP-complete, revolves around finding good approximations of the optimal solution via greedy approaches, which are very effective for non-decreasing, submodular functions (Nemhauser and Wolsey 1978).

Only recently, work on sequence submodularity has emerged. Streeter and Golovin (2009) first considered this problem in the context of online resource allocation applications. Shortly after, Alaei and Malekian (2010) introduced the term *sequence submodularity* and showed that if the submodular function is non-decreasing and differentiable, a greedy approach always achieves a solution that is at least $1 - \frac{1}{e}$ of the optimal one for the maximization problem. Zhang et al. (2016) consider *string submodularity*, which is a weaker concept as the submodularity holds for the prefix relationship instead of for any type of subsequence relationship. They improve on Alaei and Malekian's approximation by introducing additional constraints on the degree of string submodularity (*curvature*) of the objective function.

Other authors have defined sequence submodularity within a *graph*-based setting. Tschiatschek et al. (2017) consider cases in which dependencies between elements of a sequence can be captured via directed acyclic graphs (DAGs) and present an algorithm with theoretical guarantees for them. However, repetitions in the sequence are not allowed and DAG submodular functions are not necessarily string or sequence submodular. Mitrovic et al. (2018) extend this graph-based framework to graphs and hypergraphs with bounded in or out degrees.

Finally, Qian et al. (2018) take a departure from the greedy approach and propose a Pareto optimization method for sequence selection. They show that, for any class of submodular functions previously studied, their approach can always reach the best known approximation guarantee.

Against the backdrop of this body of work, we aim to show that the submodular functions appearing in practical applications do not satisfy the constraints imposed by the approaches highlighted here. However, they do present a common structure that can be exploited to equip a suitably modified greedy algorithm with strong theoretical guarantees.

Problem Statement

Let Ω be a set and $\mathbb{H}(\Omega)$ be the language over Ω, i.e. the set of sequences of elements in Ω of any length including the empty sequence $\varnothing$. Let $\mathbb{H}^d(\Omega)$ denote the sub-language consisting of all sequences in $\mathbb{H}(\Omega)$ with distinct elements. If $S = (S_1, \dots, S_n) \in \mathbb{H}(\Omega)$, we denote with $|S| = n$ the length of the sequence S. Given $R, S \in \mathbb{H}(\Omega)$, we say that R is a *subsequence* of S (denoted $R \leq S$) if R is obtained from S by eliminating some of its elements, i.e. if there exists a strictly increasing function $\rho : \{1, \dots, |R|\} \to \{1, \dots, |S|\}$ such that $R_i = S_{\rho(i)}$ for every $i = 1, \dots, |R|$. We use the following convention to indicate a specific type of subsequences: if $S = (S_1, \dots, S_n) \in \mathbb{H}(\Omega)$ and $1 \leq a \leq b \leq n$, we write $S|_a^b = (S_a, S_{a+1}, \dots, S_b)$. We put $S|_a^b = \varnothing$, if $a > b$.

In this paper, we focus on greedy algorithms for maximizing functionals defined on $\mathbb{H}(\Omega)$ that present the following *recursive* form:

$$F_g(S) = \sum_{k=1}^{n} g(S_k) \left[F(S|_1^k) - F(S|_1^{k-1}) \right] \qquad (1)$$

for $S = (S_1, \dots, S_n) \in \mathbb{H}(\Omega)$. Here, $g : \Omega \to \mathbb{R}^+$ is any function and $F : \mathbb{H}(\Omega) \to \mathbb{R}$ is a functional independent from the specific order of the elements in S, monotonic and submodular (formal definitions are given in the next section).

The problem of maximizing these functionals (typically on finite sequences with length below a given value) is significant because it lies at the heart of several practical applications, ranging from jobs scheduling to web recommendation systems, as we will see below. Note that for a general g, the functionals F_g depends on the specific ordering of the elements of the sequence, with the consequence that the classical results on set functions cannot be applied.

Application Domains

We now introduce three, very different application domains, which have a functional of the type of Eq. (1) at their core. They represent classes of problems and demonstrate the expressiveness and generality of our approach.

Search-and-Tracking (S&T) Our first domain is a state-of-the-art S&T application (Piacentini, Bernardini, and Beck 2019; Bernardini et al. 2016). S&T is the problem of locating a moving target in a given area and following it to destination. The target travels across a large geographical area by following a road network (set of paths Γ), and the observer is a UAV with imperfect sensors. When the UAV loses track of the target, a set of candidate flight search patterns Ω is selected via a Monte Carlo simulation to direct the search towards the areas in which it is more probable to rediscover the target. The UAV, however, has not enough resources to execute all candidates and a subset of patterns needs to be selected and arranged in a feasible sequence for execution.

Each pattern $\sigma \in \Omega$ provides visibility over a family of paths $\Gamma_\sigma \subseteq \Gamma$, i.e. if the target follows a route in Γ_σ, the UAV may be able to detect it while performing pattern σ. Each pattern $\sigma \in \Omega$ is also associated with: (i) a time stamp $t(\sigma)$, indicating the mid-point of a time window during which the target might plausibly be in the area covered by σ; and (ii) a detection probability ϕ_σ with the following meaning: assuming that the target has taken a route in Γ_σ, if the UAV performs the pattern σ at time $t(\sigma)$, detection will be positive with probability ϕ_σ. In all other cases, detection will be negative. An a-priori uniform probability distribution on the routes in Γ is assumed, as well as independence of the outcomes of the search experiments conditioned to the fact that the target has chosen a specified route.

Formally, this model can be described as follows. Given a sequence of patterns $S \in \mathbb{H}(\Omega)$ of length n, we consider a joint probability distribution P_S on $\{0,1\}^n \times \Gamma$ where $P_S(\omega_1, \dots, \omega_n, \gamma)$ denotes the probability that the target has taken the road γ and search in the n patterns $S_1, \dots, S_n$ have given results, respectively, $\omega_1, \dots, \omega_n$ (we assume that 1 indicates a positive detection). The probability P_S is univocally described by assuming that its marginal on Γ is the uniform distribution and that

$$P_S(\omega_1, \dots, \omega_n \,|\, \gamma) = \prod_{\substack{i\,:\,\gamma \notin \Gamma_{S_i} \\ \omega_i = 1}} (1-\omega_i) \prod_{\substack{i\,:\,\gamma \in \Gamma_{S_i} \\ \omega_i = 1}} \phi_{S_i} \prod_{\substack{i\,:\,\gamma \in \Gamma_{S_i} \\ \omega_i = 0}} (1-\phi_{S_i})$$

The S&T problem is to determine, among all the sequences of a given length, the one that minimizes the expected first detection time. To formalize this problem, we first define the following subset of sequences where patterns are ordered over time:

$$\mathcal{I} := \{S \in \mathbb{H}(\Omega) \,|\, t(S_k) \le t(S_{k+1}) \ \forall k = 1, \dots, |S|\} \qquad (2)$$

For every $S \in \mathcal{I}$ with $|S| = n$, we define the *first detection time* as the random variable $\tau_S : \{0,1\}^n \to \mathbb{R}$ such that

$$\tau_S(\omega) = \begin{cases} t(S_k) & \text{if } \omega_k = 1,\ \omega_j = 0 \text{ for } j < k \\ K & \text{if } \omega_j = 0 \text{ for every } j \end{cases}$$

where $K \ge \max_{\sigma \in \Omega} t(\sigma)$ is a constant playing the role, as we will see below, of a penalty for the fact that detection has not succeeded within S. We aim to minimize is $\mathbb{E}[\tau_S]$, i.e. the expected value of τ_S, on sequences of a fixed length T.

We now show how such goal involves the maximization of a functional of the type of Eq. (1). We define a functional $F : \mathbb{H}(\Omega) \to \mathbb{R}^+$ with $F(S) = 1 - P_S(0, \dots, 0)$ if $S \ne \emptyset$ and $F(\emptyset) = 0$. This functional represents the probability that the sequence of patterns S produces a successful detection of the target and can be efficiently computed in a recursive fashion as indicated in Piacentini et al. (2019).

Now, we can compute $\mathbb{E}[\tau_S]$ as follows:

$$\begin{aligned} \mathbb{E}[\tau_S] &= \sum_{k=1}^{|S|} t(S_k) P_{S|_1^k}(0, \dots 0, 1) + K P_S(0, \dots, 0) \\ &= \sum_{k=1}^{|S|} t(S_k) \left[F(S|_1^k) - F(S|_1^{k-1})\right] + K[1 - F(S)] \\ &= -\sum_{k=1}^{|S|} (K - t(S_k)) \left[F(S|_1^k) - F(S|_1^{k-1})\right] + K \end{aligned}$$

So, our optimization problem can be expressed as

$$\max_{\substack{S \in \mathcal{I} \\ |S| = T}} \sum_{k=1}^{T} (K - t(S_k)) \left[F(S|_1^k) - F(S|_1^{k-1})\right] \qquad (3)$$

This functional coincides with $F_g(S)$ in Eq. (1) with $g = K - t$.

Note that S&T is a particular instance of a general *detection problem* in which we consider a set Ω of possible experiments that aim to detect a specific event (e.g. the identification of a gas cloud moving through an area). A time stamp $t(\sigma) \in \mathbb{R}$ is associated with each experiment $\sigma \in \Omega$, which can be repeated multiple times (e.g., several sensors can be used at the same position). A sequence of experiments can be seen as an element of $\mathbb{H}(\Omega)$ and the theory laid out for S&T can be applied to this context too.

Job scheduling We now consider a job scheduling problem that was first studied by Stadje (1995). Assume that Ω is a set of jobs that need to be processed by a single machine subject to failure, which is modeled stochastically. We associate a number $P(\sigma)$ with each job $\sigma \in \Omega$, which represents the probability that the machine does not fail while performing σ. We assume that the machine is not aging so the probability of not failing while performing a sequence of jobs S is simply $P(S) = \prod_j P(S_j)$. Every job σ is also associated with a reward $R(\sigma)$ and a discount $d(\sigma)$ (typically, the discount depends on the time t_σ needed to complete job σ, e.g. $d(\sigma) = e^{-at_\sigma}$). The reward of performing the job σ after the sequence of jobs S has been performed is given by $d(S)R(\sigma)$, where $d(S) = \prod_j d(S_j)$. The objective function G on a sequence of jobs S is the expected total reward under the assumption that the machine keeps processing jobs of the sequence S until it fails. Formally, we have

$$G(S) = \sum_{k=1}^{|S|} P(S|_1^{k-1}) d(S|_1^{k-1}) R(S_k) \qquad (4)$$

(with the convention that $P(\emptyset)d(\emptyset) = 1$). The functional G fits the class of functionals in Eq. (1) and is formally equivalent to the functional considered in the S&T problem described above. To see this, we put $D(S) = P(S)d(S)$ and we note that, by multiplying and dividing the k-th addend in Eq. (4) by $1 - D(S_k)$, we obtain:

$$G(S) = \sum_{k=1}^{|S|} [D(S|_1^{k-1}) - D(S|_1^k)] \frac{R(S_k)}{1 - D(S_k)} \qquad (5)$$

If we now put $F(S) = 1 - D(S)$ and $g(\sigma) = \frac{R(\sigma)}{1-D(\sigma)}$, we observe that G coincides with F_g as defined in Eq. (1). The main result reported by Stadje (1995) is that, restricting G to sequences of distinct jobs of a fixed length n, the optimal solution is a sequence S for which g is decreasing, namely $g(S_1) \ge g(S_2) \ge \dots \ge g(S_n)$.

Recommender systems Finally, we present and extend a recommender system application (Ashkan et al. 2015). In this context, assume that Ω is a set of movies and the function $g : \Omega \to [0,1]$ attributes the corresponding satisfaction probability of a default user to each of them. Movies are organized under different topics, i.e. there is a set $\mathcal{T}$ of topics and a function t such that, for each $\sigma \in \Omega$, $t(\sigma) \subseteq \mathcal{T}$ is the subset of the topics covered by σ. The recommender system generates a sequence $S \in \mathbb{H}^d(\Omega)$.

The objective function $G : \mathbb{H}^d(\Omega) \to \mathbb{R}$ is the probability of the user satisfaction assuming the following stochastic model of choice: the user chooses a topic t in $\mathcal{T}$ uniformly at random and picks the first item S_i in the given sequence for which $t \in t(S_i)$. We use the notation $i(t)$ to indicate such index i. Formally, we have that $i(t) = \min\{i = 1, \dots, |S| \,|\, t \in t(S_i)\}$. The user will be satisfied with probability $g(S_{i(t)})$.

We can formally compute $G(S)$ as follows:

$$\begin{aligned} G(S) &= \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{P}(\text{satisfied} \,|\, t) \\ &= \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} g(S_{i(t)}) = \sum_{i=1}^{|S|} w(S_i) g(S_i) \end{aligned} \tag{6}$$

where $w(S_i)$ is the fraction of topics t for which S_i is the first item in the sequence that covers them. More precisely, if we define $F : \mathbb{H}^d(\Omega) \to \mathbb{R}$ so that

$$F(S) = \left| \bigcup_{i=1}^{|S|} t(S_i) \right| / |\mathcal{T}|$$

is the fraction of topics covered by the sequence S, we can write $w(S_i) = F(S|_1^i) - F(S|_1^{i-1})$, and, by substituting $w(S_i)$ in Eq. (6), we recognize that the functional is in the form of Eq. (1).

Ashkan et al. (2015) study the optimality of the functional F_g over the set of sequences of distinct items of maximal length $|\Omega|$ and discover that the solution, as in the previous example, is given by any S on which g is monotonically decreasing. In addition, they note that such optimal solution S can be trimmed by iteratively discarding all items S_i for which $F(S|_1^i) - F(S|_1^{i-1}) = 0$. In this way, they obtain the shortest possible recommended sequence of items still maximizing the satisfaction probability.

In practical applications, as also noted by the authors, it may be of interest to optimize over sequences that are not necessarily of maximal length. In this direction, we propose a generalization of the above model that also leads to a functional of the type of Eq. (1). Instead of assuming that a movie $\sigma \in \Omega$ covers a set of topics $t(\sigma)$, we associate a probability vector p^σ over $\mathcal{T}$ with each movie σ, where $p^\sigma(t)$ indicates to which extent movie σ covers topic t. Hence, we assume that the choice mechanism of the user is now the following: once the topic t has been selected, the user will pick S_1 with probability $p^{S_1}(t)$. If S_1 is not chosen (which will happen with probability $1 - p^{S_1}(t)$), the user will pick S_2 with probability $p^{S_2}(t)$ and so on. If $S_{i(t)}$ is the one chosen, the user will be satisfied with probability $g(S_{i(t)})$. In this case:

$$G(S) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{P}(\text{sat.} \,|\, t) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{i=1}^{|S|} g(S_i)\mathbb{P}(i(t) = i \,|\, t)$$

where

$$\mathbb{P}(i(t) = i \,|\, t) = (1 - p^{S_1}(t)) \cdots (1 - p^{S_{i-1}}(t)) p^{S_i}(t)$$

If we now define

$$F(S) = \sum_{i=1}^{|S|} (1 - p^{S_1}(t)) \cdots (1 - p^{S_{i-1}}(t)) p^{S_i}(t)$$

as the probability that one of the items of the sequence S is eventually picked, we have that

$$\mathbb{P}(i(t) = i \,|\, t) = F(S|_1^i) - F(S|_1^{i-1})$$

This shows that, in this more general case too, the functional G has the same structure of the functional in Eq. (1).

Sequence Submodularity

Consider the language $\mathbb{H}(\Omega)$ over a set Ω. If $S = (S_1, \dots, S_n), S' = (S'_1, \dots, S'_m) \in \mathbb{H}(\Omega)$, their concatenation is defined as:

$$S \perp S' = (S_1, \dots, S_n, S'_1, \dots, S'_m)$$

For the sake of notational simplicity, concatenations with sequences (σ) of length 1 will be denoted simply by $S \perp \sigma$ and $\sigma \perp S$, dropping the parentheses.

Definition 1. *A function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ is called* **forward / backward monotonic** *if, respectively,*

$$J(S \perp \sigma) \geq J(S), \quad J(\sigma \perp S) \geq J(S) \quad \forall S \in \mathbb{H}(\Omega), \sigma \in \Omega$$

We use instead the term **anti-monotonic** *if the inequalities are inverted.*

Definition 2. *A function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ is called* **forward / backward (sequence) submodular** *if for every $S, R \in \mathbb{H}(\Omega)$, $\sigma \in \Omega$, respectively,*

$$J(S \perp R \perp \sigma) - J(S \perp R) \leq J(S \perp \sigma) - J(S)$$
$$J(\sigma \perp R \perp S) - J(R \perp S) \leq J(\sigma \perp S) - J(S)$$

For brevity, we drop *sequence* as we are only concerned about those functions in this paper. On the subset $\Omega^n \subseteq \mathbb{H}(\Omega)$ of sequences of length exactly n, there is a natural action of the permutation group $\mathcal{S}_n$:

$$\omega = (\omega_1, \dots, \omega_n), \; \theta \in \mathcal{S}_n \longrightarrow \theta\omega := (\omega_{\theta(1)}, \dots, \omega_{\theta(n)})$$

Definition 3. *A function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ is* **permutation invariant** *if, for every $R \in \mathbb{H}(\Omega)$ and for every $\theta \in \mathcal{S}_{|R|}$, it holds $J(\theta(R)) = J(R)$.*

For permutation invariant functions, the backward and forward notions above always coincide and, in that case, we will refer to them as monotonic, anti-monotonic, and submodular functions.

A Greedy Algorithm for Sequence Functions

Let us now fix a value $T \in \mathbb{N}$ and consider the problem of maximizing a function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ on the sequences of fixed length T. A popular, simple, suboptimal algorithm for such maximization problems is the greedy algorithm by Alaei and Malekian (2010), which generalizes the classical result in Nemhauser and Wolsey (1978) to sequence functions. This algorithm produces recursively a sequence $S = (S_1, \dots, S_T)$ by adding new elements on the right side of the sequence so that, for every $k = 0, \dots, T-1$,

$$J(S|_1^k \perp S_{k+1}) \geq J(S|_1^k \perp \sigma) \quad \forall \sigma \in \Omega \tag{7}$$

Note that the first step, for $k = 0$, simply gives $J(S_1) \geq J(\sigma)$ for every $\sigma \in \Omega$; in other words, S_1 is a maximizing sequence among those of length 1.

The following result (see proof by Alaei and Malekian (2010)) gives a lower bound on the performance of the greedy algorithm in the presence of monotonicity and submodularity of J. Fix a value $T \in \mathbb{N}$, let S^T be the sequence generated by the greedy algorithm stopped at step T and $O^T \in \mathbb{H}(\Omega)$ any maximizing sequence of J restricted to sequences in $\mathbb{H}(\Omega)$ of length T.

Theorem 1. *Assume that J is backward monotonic and forward submodular. Then,*

$$J(S^T) \geq \left(1 - \frac{1}{e}\right) J(O^T) \qquad (8)$$

We remark that if J is not permutation invariant, for the result to hold, it is necessary that monotonicity is verified backward and the submodularity is verified forward. It can easily be shown that if J is forward monotonic and backward submodular, then the same estimation will hold for the backward version of the proposed greedy algorithm, where the sequence is built by adding new elements on the left side of the sequence instead of the right side.

Let us now go back to the recursive functionals of our interest, i.e. those of the type of Eq. (1). Given a permutation invariant, monotonic, and submodular function $F : \mathbb{H}(\Omega) \to \mathbb{R}$ and a function $g : \Omega \to \mathbb{R}^+$, it is simple to see that F_g is forward monotonic and forward submodular, while, in general, it does not possess the other two complementary properties. Hence, Theorem 1 cannot be applied. In Example 1, we show that the classical greedy algorithm can perform arbitrarily bad on such functionals.

Example 1. *Consider $\Omega = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ and put $\Omega_k = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ for $k \leq n$. Define $F : \mathbb{H}(\Omega) \to \mathbb{R}$ by*

$$F(S) = \begin{cases} 0 & \text{if } S = \varnothing \\ f_k & \text{if } S \in \mathbb{H}(\Omega_k) \setminus \mathbb{H}(\Omega_{k-1}) \end{cases}$$

where $f_k = 2^{k-1}$ if $k = 1, \dots, n-1$ while $f_n = 2^n$. F is permutation invariant, monotonic and submodular, since it depends only on the element in the sequence with the maximum index k. Let $g : \Omega \to \mathbb{R}^+$ be defined by $g(\sigma_k) = 2^{n-k}$. Note now that, when $S = (\sigma_k)$ is of length 1, the functional $F_g(S)$ is given by

$$F_g(\sigma_k) = g(\sigma_k)F(\sigma_k) = \begin{cases} 2^{n-1} & \text{if } k \leq n-1 \\ 2^n & \text{if } k = n \end{cases}$$

Consequently, the greedy solution S^n of length n will necessarily be such that $S^n_1 = \sigma_n$. This implies that $F(S^n|_1^k) - F(S^n|_1^{k-1}) = 0$ for every $k = 2, \dots, n$ and hence the choice of the elements S^n_k for $k \geq 2$ will have no effect on the value of the functional F_g, which will remain unchanged with value $F_g(S^n) = 2^n$. On the other hand, we can compute

$$\begin{aligned} &F_g(\sigma_1, \dots, \sigma_n) = \sum_{k=1}^{n} g(\sigma_k)[F(\sigma_1 \cdots \sigma_k) - F(\sigma_1 \cdots \sigma_{k-1})] \\ &= \sum_{k=1}^{n} 2^{n-k}(f_k - f_{k-1}) = f_n + \sum_{k=0}^{n-1} f_k(2^{n-k+1} - 2^{n-k}) \\ &= 2^n + (n-1)2^{n-1} \end{aligned}$$

Therefore,

$$\frac{F_g(\sigma_1, \sigma_2, \dots, \sigma_n)}{F_g(S^n)} = 1 + \frac{n-1}{2}$$

and, as n increases, the ratio between the two values becomes arbitrarily large. Hence, no bound of the form that is guaranteed by Theorem 1 can possibly hold in this case.

In Example 1, the optimizing sequence S is such that the corresponding values $g(S_k)$ are in decreasing order $g(S_1) \geq g(S_2) \geq \cdots$. This property is indeed general and, as such, it also characterizes the applicative problems previously introduced. It is a consequence of the following proposition:

Proposition 1. *Let $S \in \mathbb{H}(\Omega)$ and $k < |S|$ be such that $g(S_k) \leq g(S_{k+1})$. Let $\tilde{S}$ be the sequence obtained from S exchanging S_k and S_{k+1}. Then,*

$$F_g(\tilde{S}) \geq F_g(S)$$

Proof. Because of the permutation invariance of the function F, we have that:

$$\begin{aligned} &F_g(S) - F_g(\tilde{S}) \\ &= g(S_k)[F(S|_1^k) - F(S|_1^{k-1})] \\ &+ g(S_{k+1})[F(S|_1^{k+1}) - F(S|_1^k)] \\ &- g(S_{k+1})[F(S|_1^{k-1} \perp S_{k+1}) - F(S|_1^{k-1})] \\ &- g(S_k)[F(S|_1^{k+1}) - F(S|_1^{k-1} \perp S_{k+1})] \\ &= [g(S_{k+1}) - g(S_k)] \\ &\cdot [F(S|_1^{k+1}) - F(S|_1^k) - F(S|_1^{k-1} \perp S_{k+1}) + F(S|_1^{k-1})] \end{aligned}$$

The proof is concluded by observing that the last term is non-positive since $g(S_k) \leq g(S_{k+1})$ and F is submodular. □

In general, the classical greedy algorithm will not construct a sequence that presents such an ordering of the elements $g(S_1) \geq g(S_2) \geq \cdots$. This is the cause of its poor performance, as evidenced by Example 1. To overcome this difficulty, in the next section, we propose an alternative greedy algorithm where this ordering is automatically enforced.

Generalized Submodular Functions

Compared with previous work, our optimization approach is more general as it allows problems to be defined not only over $\mathbb{H}(\Omega)$ (elements can be repeated) and $\mathbb{H}^d(\Omega)$ (elements are all distinct), but also over sets in which the number of repetitions of each element can be constrained to be below a certain value. To allow for such generality, we introduce the key concept of *fully extendable* set of sequences and new notions of monotonicity and submodularity adapted to it.

Definition 4. *A subset $\mathcal{I} \subseteq \mathbb{H}(\Omega)$ is called* **fully extendable** *if the following conditions are satisfied.*

1. *For every $\sigma \in \Omega$, $(\sigma) \in \mathcal{I}$;*
2. *If $R \in \mathcal{I}$ and $Q \leq R$, then $Q \in \mathcal{I}$;*
3. *If $Q, R \in \mathcal{I}$, there exists $U \in \mathcal{I}$ such that $Q, R \leq U$ and $|U| \leq |Q| + |R|$*

The third property says that, given two sequences $Q, R \in \mathcal{I}$, there must exist another sequence $U \in \mathcal{I}$ of which both are subsequences and whose length is at most the sum of the two lengths. If Q and R do not have any element in common, the only possibility is that U is obtained by intertwining Q and R and then $|U| = |Q| + |R|$.

We denote by $\mathcal{I}(Q,R)$ the subset of sequences U satisfying property 3. defined above. Given $Q \in \mathcal{I}$, we also denote

$$\mathcal{I}^+(Q) \coloneqq \{U \in \mathcal{I} \,|\, Q \leq U,\, |U| = |Q| + 1\}$$

In other words, $\mathcal{I}^+(Q)$ consists of the sequences in $\mathcal{I}$ that are obtained from Q by adding one element. It follows from properties 1. and 3. and the considerations above that if there exist elements in Ω not appearing in Q, surely $\mathcal{I}^+(Q) \neq \emptyset$.

Example 2. $\mathbb{H}(\Omega)$ *is a fully extendable set.*

Below, we construct a family of fully extendable sets that play a crucial role in our theory. Given a function $g : \Omega \to \mathbb{R}^+$ consider an ordering $>$ of the elements of Ω for which g is non-decreasing: $\sigma > \sigma'$ implies $g(\sigma) \geq g(\sigma')$. Notice that this ordering is not unique when g is non injective. In what follows, we assume that one such ordering has been established (the specific choice made will not play any role in our future considerations). A sequence $S \in \mathbb{H}(\Omega)$ is called *g-ordered* if $S_1 \geq \cdots \geq S_{|S|}$. We fix a notation: given $S \in \mathbb{H}(\Omega)$, we denote by $n_\sigma(S)$ the number of times the element σ appears in the sequence S. The following two properties are a direct consequence of the way in which the g-ordering has been defined:

(i) Given non negative integer numbers n_σ for every $\sigma \in \Omega$, there exists exactly one g-ordered sequence S such that $n_\sigma(S) = n_\sigma$ for every $\sigma \in \Omega$.

(ii) Given two g-ordered sequences Q, R, we have that $Q \leq R$ if and only if $n_\sigma(Q) \leq n_\sigma(R)$ for every $\sigma \in \Omega$.

Given a subset $\mathcal{I} \subseteq \mathbb{H}(\Omega)$, define

$$\mathcal{I}(g) = \{S \in \mathcal{I} \mid S \text{ is } g\text{-ordered}\}$$

When $\mathcal{I} = \mathbb{H}(\Omega)$ or $\mathcal{I} = \mathbb{H}^d(\Omega)$, we will use the notation $\mathbb{H}(\Omega, g)$ and $\mathbb{H}^d(\Omega, g)$, respectively, for $\mathcal{I}(g)$. These sets characterize our running example domains: in S&T problems, the set $\mathcal{I}$ defined in Eq. (2) coincides with $\mathbb{H}(\Omega, -t)$ and represents a natural restriction of the domain to cope with time monotonicity, while, in the other two applications, optimal solutions in the literature always present this ordered structure. More general examples of fully extendable sets can be constructed as follows. For every $\sigma \in \Omega$, fix a number $n_\sigma \in \{1, 2, \dots\} \cup \{+\infty\}$ and consider the set of sequences

$$\mathcal{I} = \{S \in \mathbb{H}(\Omega) \,|\, n_\sigma(S) \leq n_\sigma \;\forall \sigma \in \Omega\} \tag{9}$$

Note that $\mathbb{H}(\Omega)$ and $\mathbb{H}^d(\Omega)$ are special cases of $\mathcal{I}$, obtained when, respectively, $n_\sigma = +\infty$ and $n_\sigma = 1$ for every $\sigma \in \Omega$.

We have the following result

Proposition 2. *Consider a function $g : \Omega \to \mathbb{R}^+$ and the set of sequences $\mathcal{I}$ defined in Eq. (9). The set $\mathcal{I}(g)$ is fully extendable.*

Proof. All singleton sequences $S = (\sigma)$ are g-ordered and respect the repetition constraint (since $n_\sigma \geq 1$). Therefore, they are in $\mathcal{I}(g)$ and property 1. in Definition 4 holds. Property 2. also holds because any subsequence Q of a sequence in $R \in \mathcal{I}(g)$ is necessarily g-ordered and satisfies, thanks to property (ii) above, the constraints $n_\sigma(Q) \leq n_\sigma(R) \leq n_\sigma$ for every $\sigma \in \Omega$. To check property 3. consider now two sequences $Q, R \in \mathcal{I}(g)$ and put, for every $\sigma \in \Omega$,

$$\bar{n}_\sigma = \max\{n_\sigma(Q), n_\sigma(R)\}$$

Let U be the only g-ordered sequence such that $n_\sigma(U) = \bar{n}_\sigma$ (see property (i)). Since by construction $\bar{n}_\sigma \leq n_\sigma$ for all $\sigma \in \Omega$, we have that $U \in \mathcal{I}(g)$. Notice now that both Q and R are subsequences of U because of property (ii). Finally,

$$|U| = \sum_{\sigma \in \Omega} n_\sigma(U) \leq \sum_{\sigma \in \Omega} [n_\sigma(Q) + n_\sigma(R)] = |Q| + |R|$$

This completes the proof. □

Let us now see the new notions of monotonicity and submodularity adapted to fully extendable sets.

Definition 5. *Given $\mathcal{I} \subseteq \mathbb{H}(\Omega)$, a function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ is called $\mathcal{I}$-**weakly monotonic** if for every $Q, R \in \mathcal{I}$, there exists $U \in \mathcal{I}(Q,R)$ such that $J(U) \geq J(Q)$.*

Definition 6. *Given $\mathcal{I} \subseteq \mathbb{H}(\Omega)$, a function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ is called $\mathcal{I}$-**strongly forward submodular** if for every $Q, R, U \in \mathcal{I}$ with $Q \leq R$ and $\sigma \in \Omega$ such that $R \perp \sigma \perp U \in \mathcal{I}$, it holds, $J(R \perp \sigma \perp U) - J(R \perp U) \leq J(Q \perp \sigma \perp U) - J(Q \perp U)$.*

Whenever, $\mathcal{I} = \mathbb{H}(\Omega)$, we will simply talk of weakly monotonic and strongly forward submodular functions.

Remark 1. *When $\mathcal{I} = \mathbb{H}(\Omega)$, since $Q \perp R, R \perp Q \in \mathcal{I}(Q,R)$, any backward or forward monotonic function is also weakly monotonic. Moreover, a strongly forward submodular function is also forward submodular. This motivates the terms weak and strong that we use.*

A New Greedy Algorithm

We now introduce a generalized greedy algorithm and show that, for functionals that are $\mathcal{I}$-weakly monotonic and $\mathcal{I}$-strongly forward submodular, this new algorithm ensures the same performance guaranteed by Theorem 1. We will then show how to apply this result to our problems.

Take a function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ and a fully extendable set $\mathcal{I} \subseteq \mathbb{H}(\Omega)$. We fix a value $T \in \mathbb{N}$: the goal is to maximize J over the subset of $\mathcal{I}$ of the sequences of length T. Put O^T to be any such maximizing sequence for J.

We now consider a variation of the greedy algorithm to approximately solve this maximization problem. The algorithm produces recursively an ordered sequence $S^T = (S_1^T, \dots, S_T^T) \in \mathcal{I}$ in the following way:

- $S^1 = (S_1^1)$ where $S_1^1 \in \operatorname{argmax}_{\sigma \in \Omega} J(\sigma)$;
- Given $S^k = (S_1^k, \dots, S_k^k) \in \mathcal{I}$, we define

$$S^{k+1} = \operatorname*{argmax}_{U \in \mathcal{I}^+(S^k)} J(U)$$

In other words, instead of simply augmenting the sequence on the right hand side as the traditional greedy algorithm does, we allow each new element to be placed in any position among the elements of the previous sequence.

The following result provides a similar bound to that of Theorem 1.

Theorem 2. *Consider a function $J : \mathbb{H}(\Omega) \to \mathbb{R}$ and a fully extendable set $\mathcal{I} \subseteq \mathbb{H}(\Omega)$ and assume that J is $\mathcal{I}$-weakly monotonic and $\mathcal{I}$-strongly forward submodular. Let O^T be a maximizing sequence for J among the sequences in $\mathcal{I}$ of length T and let S^T be the result of the previous algorithm. Then,*

$$J(S^T) \geq \left(1 - \frac{1}{e}\right) J(O^T)$$

Proof. For simplicity of notation, in the proof, we put $O = O^T$. Fix $k < T$ and consider

$$\Lambda = (\lambda_1, \dots, \lambda_n) \in \operatorname*{argmax}_{U \in \mathcal{I}(S^k, O)} J(U)$$

We consider a partition of the indices

$$\{1, 2, \dots, n\} = \{i_1, i_2, \dots, i_k\} \cup \{j_1, j_2, \dots, j_m\}$$

where $i_1 < i_2 < \cdots < i_k$ are such that $S^k_l = \lambda_{i_l}$ for $l = 1, \dots, k$ and $j_1 < j_2 < \cdots < j_m$ with $m = n - k$ are the remaining indices.

We now consider, for $0 \leq t \leq m$, the sequence $\Lambda^{(t)}$ obtained from Λ by removing the elements $\lambda_{j_m}, \lambda_{j_{m-1}}, \dots, \lambda_{j_{t+1}}$. Note that, by property 2. of fully extendable sets, $\Lambda^{(t)} \in \mathcal{I}$ for every t and that $\Lambda^{(m)} = \Lambda$ and $\Lambda^{(0)} = S^k$. We can write

$$J(\Lambda) - J(S^k) = \sum_{t=1}^{m} \left[J(\Lambda^{(t)}) - J(\Lambda^{(t-1)}) \right] \tag{10}$$

Using the property of $\mathcal{I}$-strongly forward submodularity and removing the elements $\lambda_{j_{t-1}}, \dots, \lambda_{j_0}$ from $\Lambda^{(t-1)}$ and $\Lambda^{(t)}$, we obtain that

$$J(\Lambda^{(t)}) - J(\Lambda^{(t-1)}) \leq J(U^{(t)}) - J(S^k) \tag{11}$$

for some $U^{(t)} \in \mathcal{I}^+(S^k)$ (a sequence obtained from S^k adding in some position the element λ_{j_t}). Given the definition of the extended greedy solution S^k, it follows that $J(U^{(t)}) \leq J(S^{k+1})$. This fact together with Eqs. (10) and (11) yields:

$$J(\Lambda) - J(S^k) \leq T \left[J(S^{k+1}) - J(S^k) \right] \tag{12}$$

The assumption of $\mathcal{I}$-weak monotonicity and the choice of Λ to maximize J on $\mathcal{I}(S^k, O)$ ensure that $J(\Lambda) \geq J(O)$. Using this fact inside Eq. (12) gives:

$$J(S^{k+1}) \geq \frac{1}{T} J(O) + \left(1 - \frac{1}{T}\right) J(S^k)$$

for every $k = 0, \dots, T-1$. Applying recursively this relation, we obtain that

$$\begin{aligned} J(S^T) &\geq \tfrac{1}{T} \sum_{i=0}^{T-1} \left(1 - \tfrac{1}{T}\right)^i J(O) \\ &= \left[1 - \left(1 - \tfrac{1}{T}\right)^T\right] J(O) \geq \left(1 - \tfrac{1}{e}\right) J(O) \end{aligned}$$

□

Solving the Application Problems

We now go back to our optimization problem on functionals of the type of Eq. (1) and study under which conditions we can apply the theory laid out in the previous section. We then discuss these results for our applicative domains.

We fix a permutation invariant, monotonic and submodular functional F, a function $g : \Omega \to \mathbb{R}^+$, and we consider the functional F_g as defined in Eq. (1). We also fix a permutation invariant set $\mathcal{I} \subseteq \mathbb{H}(\Omega)$ such that $\mathcal{I}(g)$ is fully extendable. We aim to maximize $F_g(S)$ over the sequences in $\mathcal{I}$ of a given length T. Proposition 1 implies that we can always restrict to g-ordered sequences. Formally, it holds

Proposition 3. *Given a permutation invariant set $\mathcal{I} \subseteq \mathbb{H}(\Omega)$, it holds that, for every $T \in \mathbb{N}$,*

$$\max_{\substack{S \in \mathcal{I} \\ |S| = T}} F_g(S) = \max_{\substack{S \in \mathcal{I}(g) \\ |S| = T}} F_g(S) \tag{13}$$

Note that the optimization problem in the S&T application is directly defined on the set $\mathcal{I}(g)$. Instead, in the other two applicative domains, the authors define the problems on the set $\mathcal{I} = \mathbb{H}(\Omega)^d$ and show that relation (13) hold. Proposition 3 establishes this equivalence as a general fact valid for all these optimization problems.

Remark 2. *If we maximize over $\mathcal{I} = \mathbb{H}^d(\Omega)$ with sequences of maximal length $T = |\Omega|$, Proposition 1 yields the stronger result that each element in $\mathcal{I}(g)$ is actually a maximum. This result has been proven by Stadje (1995) and Ashkan et al. (2015) for the specific application scenarios that they tackle.*

Our aim is now to show that, under suitable assumptions, the functional F_g satisfies the assumptions of Theorem 2 with respect to the fully extendable set $\mathcal{I}(g)$.

We set the following notation:

$$\begin{aligned} &\Delta F(R, \sigma) = F(R \perp \sigma) - F(R) \\ &\Delta^2 F(R, \sigma_1, \sigma_2) = \Delta F(R \perp \sigma_1, \sigma_2) - \Delta F(R, \sigma_2) \\ &= F(R \perp \sigma_1 \perp \sigma_2) - F(R \perp \sigma_1) - F(R \perp \sigma_2) + F(R) \end{aligned} \tag{14}$$

Note that the submodularity of F is equivalent to the requirement that $\Delta F(R, \sigma)$ is antimonotonic in R for every fixed $\sigma \in \Omega$. The following is the main result of this section. It guarantees that the functional F_g is $\mathcal{I}(g)$-weakly monotonic and $\mathcal{I}(g)$-strongly forward submodular if F, beyond being monotonic and submodular, also satisfies a property concerning the second variation term defined in Eq. (14).

Theorem 3. *If F is a permutation invariant, monotonic and submodular function that also satisfies the property*

$$\Delta^2 F(R, \sigma_1, \sigma_2) \text{ is monotonic in } R \text{ for every } \sigma_1, \sigma_2 \tag{15}$$

then, for any non-negative function g and permutation invariant set $\mathcal{I} \subseteq \mathbb{H}(\Omega)$ such that $\mathcal{I}(g)$ is fully extendable, it holds that F_g is $\mathcal{I}(g)$-weakly monotonic and $\mathcal{I}(g)$-strongly forward submodular.

Proof. We first show that F_g is $\mathcal{I}(g)$-weakly monotonic by proving the stronger result that, given $Q \leq U \in \mathcal{I}(g)$, it holds $J(U) \geq J(Q)$. By induction, it is sufficient to prove that in the special case when $|U| = |Q| + 1$. In this case, we have that

$Q = R \perp S$ and $U = R \perp \sigma \perp S$ for suitable sequences R, S and $\sigma \in \Omega$. We reason as follows:

$$\begin{aligned} & F_g(R \perp \sigma \perp S) - F_g(R \perp S) \\ & = g(\sigma) \Delta F(R, \sigma) \\ & + \sum_{k=1}^{|S|} g(S_k) \left[\Delta F(R \perp \sigma \perp S|_1^{k-1}, S_k) - \Delta F(R \perp S|_1^{k-1}, S_k) \right] \end{aligned} \tag{16}$$

Note now that

$$\begin{aligned} & \sum_{k=1}^{|S|} \left[\Delta F(R \perp \sigma \perp S|_1^{k-1}, S_k) - \Delta F(R \perp S|_1^{k-1}, S_k) \right] \\ & = F(R \perp \sigma \perp S) - F(R \perp \sigma) - F(R \perp S) + F(R) \\ & = \Delta F(R \perp S, \sigma) - \Delta F(R, \sigma) \geq -\Delta F(R, \sigma) \end{aligned} \tag{17}$$

where the last inequality follows from the fact that, since F is monotonic, $\Delta F(R \perp S, \sigma) \geq 0$.

Using inequality (17) in (16), we obtain

$$\begin{aligned} & F_g(R \perp \sigma \perp S) - F_g(R \perp S) \\ & \geq \sum_{k=1}^{|S|} [g(S_k) - g(\sigma)] \\ & \cdot \left[\Delta F(R \perp \sigma \perp S|_1^{k-1}, S_k) - \Delta F(R \perp S|_1^{k-1}, S_k) \right] \geq 0 \end{aligned}$$

The last inequality follows from the fact that $g(S_k) - g(\sigma) \leq 0$ for every k and F being submodular. This proves that F_g is $\mathcal{I}(g)$-weakly monotonic.

We now investigate submodularity. We choose $Q, R, U \in \mathcal{I}(g)$ with $Q \leq R$ and $\sigma \in \Omega$ such that $R \perp \sigma \perp U \in \mathcal{I}(g)$. By following the same computation as in (16) and using the definition of $\Delta^2 F$, we obtain that

$$\begin{aligned} & [F_g(R \perp \sigma \perp U) - F_g(R \perp U)] \\ & -[F_g(Q \perp \sigma \perp U) - F_g(Q \perp U)] \\ & = g(\sigma)[\Delta F(R, \sigma) - \Delta F(Q, \sigma)] + \sum_{k=1}^{|U|} g(U_k) \\ & \cdot \left[\Delta^2 F(R \perp U|_1^{k-1}, \sigma, U_k) - \Delta^2 F(Q \perp U|_1^{k-1}, \sigma, U_k) \right] \end{aligned} \tag{18}$$

By rewriting $\Delta^2 F$ functions as differences of ΔF functions and repeating the computation done in (17), we obtain that

$$\begin{aligned} & \sum_{k=1}^{|U|} \left[\Delta^2 F(R \perp U|_1^{k-1}, \sigma, U_k) - \Delta^2 F(Q \perp U|_1^{k-1}, \sigma, U_k) \right] \\ & = \Delta F(R \perp U, \sigma) - \Delta F(R, \sigma) \\ & -\Delta F(Q \perp U, \sigma) + \Delta F(Q, \sigma) \\ & \leq -[\Delta F(R, \sigma) - \Delta F(Q, \sigma)] \end{aligned} \tag{19}$$

where the last inequality follows from the fact that, since F is submodular, $\Delta F(R \perp U, \sigma) - \Delta F(Q \perp U, \sigma) \leq 0$.

Using inequality (19) in (18), we have

$$\begin{aligned} & [F_g(R \perp \sigma \perp U) - F_g(R \perp U)] \\ & -[F_g(Q \perp \sigma \perp U) - F_g(Q \perp U] \\ & \leq \sum_{k=1}^{|U|} [g(U_k) - g(\sigma)] \\ & \cdot \left[\Delta^2 F(R \perp U|_1^{k-1}, \sigma, U_k) - \Delta^2 F(Q \perp U|_1^{k-1}, \sigma, U_k) \right] \leq 0 \end{aligned}$$

The last inequality follows from the fact that $g(U_k) - g(\sigma) \leq 0$ for every k and from the monotonicity assumption on $\Delta^2 F$. This proves that F_g is $\mathcal{I}(g)$-strongly submodular. □

Application Domains

To apply Theorem 3 to our applicative contexts, we need to verify, for each of them, that the corresponding functional F satisfies the following properties:

- F is permutation invariant;
- F is monotonic;
- F is submodular (equivalently, $\Delta F(R, \sigma)$ is anti-monotonic in R for any fixed σ);
- $\Delta^2 F(R, \sigma_1, \sigma_2)$ is monotonic in R for any fixed σ_1, σ_2.

Let us start with the **S&T** application. In that context, $F(S)$ is the probability that the target is detected thanks to the execution of the sequence of search patterns S. Given the assumptions made on the probabilistic model, $F(S)$ does not depend on the order of disclosing the outcome of the searches, equivalently, it is permutation invariant. The other properties can also be derived from the probabilistic interpretation. Specifically, monotonicity simply follows from the fact that the larger the set of search patterns performed, the larger is the probability of positive detection. Note now that $\Delta F(R, \sigma)$ is the probability that detection is negative in R and successful in σ: as a consequence, the larger is R and the smaller is $\Delta F(R, \sigma)$. This says that $\Delta F(R, \sigma)$ is anti-monotonic in R. Similarly, $-\Delta^2 F(R, \sigma_1, \sigma_2)$ is the probability that detection is negative in R and successful both in σ_1 and in σ_2 and, for the same reasons, it is anti-monotonic in R. Therefore, $\Delta^2 F(R, \sigma_1, \sigma_2)$ is monotonic in R.

Regarding the **job scheduling** domain, note that $D(S)$ is permutation invariant and anti-monotonic in S and observe

$$\Delta D(S, \sigma) = D(S \perp \sigma) - D(S) = D(S)[D(\sigma) - 1]$$

$$\begin{aligned} & \Delta^2 D(S, \sigma_1, \sigma_2) \\ & = D(S \perp \sigma_1 \perp \sigma_2) - D(S \perp \sigma_1) - D(S \perp \sigma_2) + D(S) \\ & = D(S)[1 - D(\sigma_1)][1 - D(\sigma_2)] \end{aligned}$$

This implies (since $D(\sigma) \leq 1$ for every σ) that $\Delta D(S, \sigma)$ is monotonic in S for every σ, while $\Delta^2 D(S, \sigma_1, \sigma_2)$ is anti-monotonic in S for every σ_1 and σ_2. Considering the complementary function $F(S) = 1 - D(S)$, we immediately obtain that F satisfies the assumptions of Theorem 3.

Finally, for the **recommender systems** case, the function $F(S)$ is the probability that an item has been picked within the sequence S and is thus, mathematically, identical to the probability of the S&T problem example. This implies that Theorem 3 is applicable to this case too.

Experimental Results

To show the potential of our method, we now provide explicit numerical simulations for the S&T application described above. In particular, we now show the advantage of using the generalized greedy algorithm over the standard one by running both algorithms on several, randomly generated S&T problems. To highlight when the two algorithms exhibit different behaviors, we consider scenarios in which the detection probability of each pattern depends on the execution time associated with it. If the patterns associated with a lower t have a high detection probability, the standard and

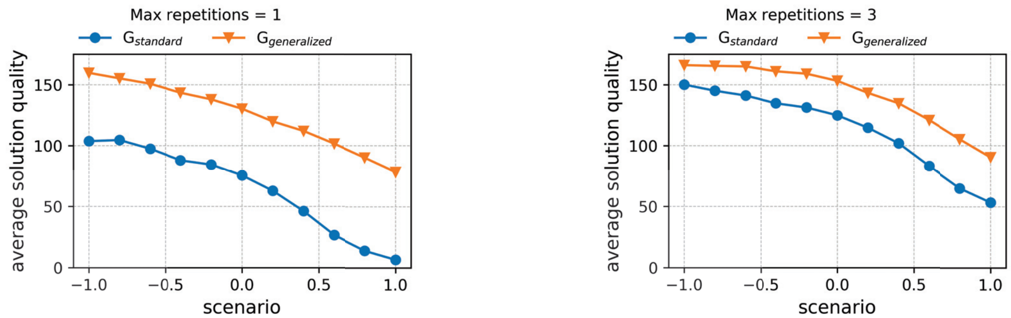

Figure 1: Average objective values obtained by the standard and the generalized greedy algorithms.

the generalized greedy search perform similarly. They prefer these early patterns by placing them at the beginning of the sequence and add the remaining patterns to the end of the sequence. Conversely, if the detection probabilities associated with patterns with a greater t are high enough, the standard greedy immediately places those patterns at the beginning of the sequence, but, as patterns are only added on the right side of the sequence, it never exploits early search patterns. In this way, it constructs short sequences that do not make full advantage of the richness of the set Ω. Instead, the generalized greedy, being free to place patterns in any position, manages to exploit both types of patterns.

We generate 11,000 realistic problems instances, each with 20 candidate patterns and 40 destinations.[1] Each pattern σ is associated with a random sample of destinations. Time stamps are generated sequentially by taking a random sequence of all the search patterns $\bar{S} = (\sigma_1, \ldots, \sigma_n)$ and imposing that a search pattern $t(\sigma_i) = t(\sigma_{i-1}) + r$, where r is a random number. The detection probability is a linear function of the indexes of the sequence of search patterns in $\bar{S}$ with different angular coefficients: $\phi_{\sigma_i} = m \cdot i + q$, where m is a value between -1 and 1, and q is such that $\sum_i \phi_{\sigma_i}$ is constant across the scenarios. A scenario with $m = -1$ corresponds to the case of patterns with a lower time stamp having higher detection probabilities, while, a scenario with $m = 1$, represent the case of patterns with higher time stamp having higher detection probabilities. When $m = 0$, all the patterns have the same detection probability.

For each problem, we run the generalized and the standard greedy algorithms over sequences of maximal length 10. We consider two types of sequences: in the first, we establish that all patterns must be distinct, in the second, instead, we allow at most 3 repetitions for each pattern.

Figure 1 shows the average objective values found by the two algorithms for different scenarios: the left plot corresponds to the case of distinct patterns, while the right plot to the case of a maximum of 3 repetitions per pattern. The figure shows that, in all cases, the generalized greedy algorithm dominates the standard algorithm. As expected, the difference in performance is particularly high (considering the ratio) in scenarios where the search patterns associated with a greater execution time have a higher detection probability. The average running time across all scenarios of the two algorithms and across all instances is reported in Table 1. While the generalized greedy algorithm is slightly more time consuming than the standard algorithm, the runtime is acceptable the real applications as the optimization of the objective function is typically performed within a time limit of one minute (Piacentini, Bernardini, and Beck 2019; Bernardini, Fox, and Long 2017).

Table 1: Average running time (in seconds) of the standard and the generalized greedy algorithms with a maximum of 1 or 3 pattern repetitions.

	max rep = 1	max rep = 3
$G_{standard}$	0.004 ± 0.001	0.015 ± 0.004
$G_{generalized}$	0.016 ± 0.006	0.067 ± 0.025

Conclusions

In this paper, we show that, in several applicative domains, the problem of finding a sequence of objects that maximizes a reward can be expressed as the maximization of a recursive functional that exhibits the structure captured by Eq. (1). After proving that existing greedy algorithms do not yield strong theoretical guarantees for such a functional, we study its properties and introduce the concepts of weak monotonicity and strong forward\backward submodularity. By leveraging them, we introduce an efficient generalized greedy approach that ensures finding solutions that are $O(1 - \frac{1}{e})$ of the optimal. Our method is general and can be applied to any domain with an objective function that can be transformed in the form of Eq. (1). To support this thesis, we present evidence that our technique works across several applications and provide explicit numerical simulations for one particular domain, S&T. The experiments directly show the power of our new algorithm. Our work contributes to the discussion on submodularity by stepping away from the specific details of practical applications and presenting general properties of functionals often encountered in them, which can be exploited to find better solutions more efficiently.

[1]For details, see supplementary material available online at https://doi.org/10.5281/zenodo.3695080

Acknowledgements

This work has been supported by EPSRC Grant EP/S016473/1 and by MIUR Grant "Dipartimenti di Eccellenza 2018-2022" (CUP: E11G18000350001).

References

Alaei, S., and Malekian, A. 2010. Maximizing sequence-submodular functions and its application to online advertising. In *arXiv:1009.4153v1*, 1–18.

Ashkan, A.; Kveton, B.; Berkovsky, S.; and Wen, Z. 2015. Optimal greedy diversity for recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, 1742–1748. AAAI Press.

Bernardini, S.; Fox, M.; Long, D.; and Piacentini, C. 2016. Leveraging probabilistic reasoning in deterministic planning for large-scale autonomous search-and-tracking. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS)*.

Bernardini, S.; Fox, M.; and Long, D. 2017. Combining temporal planning with probabilistic reasoning for autonomous surveillance missions. *Autonomous Robots* 41(1):181–203.

Dughmi, S.; Roughgarden, T.; and Sundararajan, M. 2012. Revenue submodularity. *Theory of Computing* 8:95–119.

Fujishige, S. 2005. *Submodular Functions and Optimization*. Annals of Discrete Mathematics, Elsevier.

Golovin, D., and Krause, A. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* 42(1):427–486.

Hoi, S. C. H.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, 417–424. New York, NY, USA: ACM.

Krause, A., and Guestrin, C. 2005. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, 324–331. Arlington, Virginia, United States: AUAI Press.

Krause, A., and Guestrin, C. 2007. Near-optimal observation selection using submodular functions. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, 1650–1654. AAAI Press.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9:235–284.

Lehmann, B.; Lehmann, D.; and Nisan, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55(2):270 – 296.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 785–794. New York, NY, USA: ACM.

Mitrovic, M.; Feldman, M.; Krause, A.; and Karbasi, A. 2018. Submodularity on hypergraphs: From sets to sequences. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 1177–1184.

Nemhauser, G. L., and Wolsey, L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14:265–294.

Piacentini, C.; Bernardini, S.; and Beck, C. 2019. Autonomous Target Search with Multiple Coordinated UAVs. *Journal of Artificial Intelligence Research* 65:519–568.

Qian, C.; Feng, C.; and Tang, K. 2018. Sequence selection by pareto optimization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 1485–1491. International Joint Conferences on Artificial Intelligence Organization.

Stadje, W. 1995. Selecting jobs for scheduling on a machine subject to failure. *Discrete Applied Mathematics* 63(3):257 – 265.

Streeter, M., and Golovin, D. 2009. An online algorithm for maximizing submodular functions. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc. 1577–1584.

Tschiatschek, S.; Singla, A.; and Krause, A. 2017. Selecting sequences of items via submodular maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 2667–2673. AAAI Press.

Zhang, Z.; Chong, E. K. P.; Pezeshki, A.; and Moran, W. 2016. String submodular functions with curvature constraints. *IEEE Transactions on Automatic Control* 61(3):601–616.