

On certain greedoid polyhedra, partially indexable scheduling
problems, and extended restless bandit allocation indices
Submitted to *Mathematical Programming*

José Niño-Mora*

April 5, 2000

*Dept. of Economics and Business, Universitat Pompeu Fabra, E-08005 Barcelona, Spain. E-mail: jose.nino-mora@econ.upf.es. Web: www.econ.upf.es/~ninomora. This research was partially supported by CICYT Grant TAP98-0229, and by UPF Grant COFREA99-006.

Abstract

We present a polyhedral framework for establishing general structural properties on optimal solutions of stochastic scheduling problems, where multiple job classes vie for service resources: the existence of an optimal priority policy in a given family, characterized by a *greedoid* (whose feasible class subsets may receive higher priority), where optimal priorities are determined by class-ranking indices, under *restricted* linear performance objectives (*partial indexability*). This framework extends that of Bertsimas and Niño-Mora (1996), which explained the optimality of priority-index policies under *all* linear objectives (*general indexability*). We show that, if performance measures satisfy *partial conservation laws* (with respect to the greedoid), which extend previous generalized conservation laws, then the problem admits a strong LP relaxation over a so-called *extended greedoid polytope*, which has strong structural and algorithmic properties. We present an adaptive-greedy algorithm (which extends Klimov's) taking as input the linear objective coefficients, which (1) determines whether the optimal LP solution is achievable by a policy in the given family; and (2) if so, computes a set of class-ranking indices that characterize optimal priority policies in the family. In the special case of project scheduling, we show that, under additional conditions, the optimal indices can be computed separately for each project (*index decomposition*). We further apply the framework to the important restless bandit model (two-action Markov decision chains), obtaining new index policies, that extend Whittle's (1988), and simple sufficient conditions for their validity. These results highlight the power of polyhedral methods (the so-called *achievable region approach*) in dynamic and stochastic optimization.

Key words: stochastic scheduling, restless bandits, greedoids, polyhedral methods, conservation laws, achievable region.

Journal of Economic Literature Classification: C60, C61.

1 Introduction

The application of polyhedral methods in the problem domains of combinatorial optimization, and of dynamic and stochastic optimization, shows remarkably disparate degrees of development. In combinatorial optimization, the polyhedral approach has been established as a powerful methodology for designing and analyzing both optimal and heuristic solutions, spanning the major research area of *polyhedral combinatorics*. In contrast, the use of polyhedral (or, more generally, mathematical programming) methods in dynamic and stochastic optimization, which has been called the *achievable region approach* (cf. [6]), is substantially less developed. This is one among several emerging methodologies that aim to overcome the limitations of the standard solution approach in that domain, based on dynamic programming (DP): DP formulations, hindered by the *curse of dimensionality*, have not yielded a widely applicable methodology for solving, exactly or approximately, many important models motivated by modern applications.

The development of the achievable region approach has mainly concentrated in the area of *stochastic scheduling*. Such problems are concerned with the optimal dynamic allocation of servers/machines to jobs of multiple classes subject to random dynamics and processing requirements. The goal is to design a *scheduling policy*, belonging in a typically large space of *admissible* policies, which specifies dynamically how jobs vying for a server's attention are to be prioritized for service, with the goal of optimizing a given performance objective (e.g., minimizing the class-weighted time-average job delay).

Faced with a stochastic scheduling problem, the achievable region approach proceeds in three steps: (1) the first step involves identifying the relevant vector of performance measures (e.g., the time-average delay for each job class), and characterizing the corresponding *achievable region*, spanned by achievable performance vectors under all admissible policies; the characterization should be in the form of tractable mathematical programming – preferably linear programming (LP) – constraints; if a full characterization is not available, a tractable *relaxation* is sought instead; this yields a deterministic mathematical programming formulation (exact or relaxed) of the given stochastic scheduling problem, which involves optimizing the performance objective over such constraints; (2) the second step involves solving the mathematical programming formulation, either through general methods, or, preferably, through a specific algorithm that exploits special structure; and (3) the third step involves designing a scheduling policy out of the information provided by the optimal solution to the mathematical program; such policy can be either optimal or heuristic; in the latter case, that solution provides bounds on the extent of the policy's suboptimality.

This three-step plan has been successfully deployed in an increasingly general range of stochastic scheduling models, starting with the pioneering work in [5]. That paper addressed the problem of scheduling a single server on a multiclass $M/M/1$ queue to minimize a time-average class-weighted delay objective. The achievable region of mean delays for each job class was characterized as a polyhedron with strong structural and algorithmic properties: a *polymatroid*, introduced in [7], related to a *matroid* over *all* job class subsets, whose vertices are achieved by priority policies. The characterization followed from the satisfaction by the performance vector of mean delays of a set of *work conservation laws*. The optimality of the classical $c\mu$ -rule, which computes an index for each job class, and then assigns higher priority to classes with larger indices, was thus shown to follow from the optimality of the *greedy algorithm* for solving an LP over a polymatroid. Increasingly general frameworks for establishing the optimality of such *priority-index policies* were developed in [8], [18], and [1]. The latter paper advanced beyond the polymatroidal structure, establishing that

if performance measures satisfy *generalized conservation laws*, then the achievable region is an *extended polymatroid* (introduced in [19]), defined by linear constraints associated with *all* class subsets, whose vertices are also achieved by priority policies. The optimal priority indices corresponding to an arbitrary linear objective are efficiently computed by an *adaptive-greedy algorithm* introduced by Klimov in [11]. This framework provided a new polyhedral explanation for the optimality of the Gittins index policy in classical multiarmed bandits (see [10]), and several extensions. Note that all such polyhedral frameworks identify conditions on a scheduling problem's performance measures under which the problem exhibits a *general indexability* property: an optimal set of priority indices can be computed for *each* linear performance objective.

In many important models, however, satisfaction of general indexability is too strong a requirement. The relevant concern is instead to establish the optimality of a given restricted family of priority policies, where optimal priorities are determined by ranking indices as above, for an appropriately restricted range of linear performance objectives. We call such property *partial indexability*.

A major example where partial indexability, as opposed to general indexability, is the relevant concern, is given by the *restless bandit problem*, introduced by Whittle in [21]. This is arguably the most promising extension of the classical multiarmed bandit model in stochastic scheduling. It involves a given number of stochastic projects (or *restless bandits*), modeled as general Markov decision chains (MDC) having two actions (active and passive) available in each state. At each discrete time epoch, a fixed number of projects must be engaged (active). The problem consists in designing a scheduling policy, which selects the projects to be engaged at each time, in order to optimize a total expected discounted reward/cost objective. Unlike the classical model, the restless bandit problem is computationally intractable (cf. [16]). Yet, in a pioneering analysis, Whittle [21] presented a priority-index heuristic policy for the problem, having good asymptotic properties (see [20]), which is motivated by the optimal solution of a relaxed problem. The Whittle indices (associated with project states) are computed separately for each project, as they characterize the optimal solutions to a parametric family of single-project subproblems. Unlike the Gittins indices (which they extend), the Whittle indices are typically *not* defined for *all* reward/cost objective coefficients, but only, possibly, for *some* of them. This motivated the development of alternative index policies in [4]. Furthermore, the index definition given by Whittle does not provide a qualitative understanding of the conditions under which the indices are well defined. As Whittle stated in [21],

... one would very much like to have simple sufficient conditions for indexability. At the moment, none are known.

Motivated by the above considerations, we present in this paper a general polyhedral framework for establishing the optimality of a given, restricted family of priority policies in a stochastic scheduling problem, where the optimal priorities are determined by ranking indices, under correspondingly restricted linear objectives (partial indexability). The given policy family is characterized by the class subsets that may receive higher priority, which are assumed to form a *greedoid* (cf. [12]). We show that, if system performance measures satisfy a set of *partial conservation laws* (which generalize the extended conservation laws in [1]), associated with the given greedoid, then the problem admits an LP relaxation over a certain polyhedron, which we call an *extended greedoid polytope*. Such polyhedra possess, like the extended polymatroids (cf. [1]) which they generalize, strong structural and algorithmic properties.

We present an extension of Klimov’s [11] adaptive-greedy algorithm, adapted to the greedoid structure, which takes as input the vector of linear objective coefficients. The algorithm’s output (1) determines whether the corresponding optimal LP solution is achievable by a priority policy in the given family; and (2) if so, computes a set of ranking indices that characterize optimal priority policies in the family. In the case where the scheduling problem corresponds to a *project scheduling* model, we identify additional conditions under which the indices possess a strong *decomposition property*, which allows their independent computation for each project (e.g., in parallel). We further apply the framework to the restless bandit model, presenting new definitions of priority indices, which extend Whittle’s, together with sufficient conditions for their validity.

The paper is organized as follows. Section 2 describes a motivating problem, which involves establishing structural properties on optimal solutions of a single restless bandit (two-action Markov decision chains). Section 3 introduces the notion of extended greedoid polytope, and develops structural and algorithmic properties for solving LP problems over such polyhedra. Section 4 presents the polyhedral framework of partial conservation laws, for establishing the partial indexability property in stochastic scheduling problems. Section 5 applies the framework to the motivating single restless bandit model of Section 2, presenting new definitions of priority indices, which extend those given by Whittle [21] and simple sufficient conditions for their validity. Finally, Section 6 illustrates the framework through a specific single restless bandit example.

The results presented in this paper substantially extend those developed in [13]: in that paper, we introduced the notion of partial conservation laws, and applied it to analyze the indexability property of restless bandits introduced by Whittle, which is a special case of that introduced in this paper. No greedoid structure was considered in that paper.

In the companion paper [15] we have developed a similar polyhedral framework, also based on an underlying greedoid structure, for designing and analyzing *heuristic* index policies with given structural properties.

2 Motivating problem: structured optimal solutions for single restless bandit problems

2.1 Model formulation

Consider a single *restless bandit*, i.e., a two-action Markov decision chain (MDC) (cf. [17]), whose state $X(t)$ evolves over discrete time epochs $t \geq 0$ through a finite state space N . The process $\{X(t) : t \geq 0\}$ is controlled by a *policy* u , which decides at each time t which of *two actions* available, *passive* ($a(t) = 0$) or *active* ($a(t) = 1$), is to be taken. To be *admissible*, a policy is only required to belong in the class \mathcal{U} of *nonanticipative* policies, which base decisions on the history of states visited up to and including the current time. Taking action $a \in \{0, 1\}$ in state $e \in N$ has two direct effects: first, an immediate cost C_e^a is incurred; second, the state at the next time epoch is determined through a Markovian transition rule, being $e' \in N$ with probability $p_{ee'}^a$. Costs are discounted in time by factor $0 < \beta < 1$. The initial state $X(0)$ is known up to a probability distribution, being e with probability p_e , for $e \in N$. We write $\mathbf{C}^a = (C_e^a)_{e \in N}$, $\mathbb{P}^a = (p_{ee'}^a)_{e, e' \in N}$, for $a \in \{0, 1\}$, and $\mathbf{p} = (p_e)_{e \in N}$.

Let us denote by $Z^u(\mathbf{C}^0, \mathbf{C}^1)$ the total expected net present value of costs incurred over

an infinite horizon under policy $u \in \mathcal{U}$, i.e.,

$$Z^u(\mathbf{C}^0, \mathbf{C}^1) = E_u \left[\sum_{t=0}^{\infty} C_{X(t)}^{a(t)} \beta^t \right]. \quad (1)$$

In (1), $E_u[\cdot]$ represents the expectation operator under policy u , conditional on initial state distribution \mathbf{p} . The corresponding *single restless bandit problem* consists in finding a policy $u^{\text{OPT}} \in \mathcal{U}$ that minimizes performance objective $Z^u(\mathbf{C}^0, \mathbf{C}^1)$, and in computing the corresponding optimum value $Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1)$:

$$Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1) = \min \{ Z^u(\mathbf{C}^0, \mathbf{C}^1) : u \in \mathcal{U} \}. \quad (2)$$

It is well known from DP theory (cf. [17]), that there exists an optimal policy to (2) that is (i) *stationary* and *deterministic*, i.e., it assigns a fixed action to each state; and (ii) optimal for *any* initial state probability vector \mathbf{p} .

Note that each stationary deterministic policy is characterized by the set of states where it takes the active action. Let us thus associate with each state space subset $S \in 2^N$ the corresponding *S-active policy*: this takes the active action over states in S , and the passive action otherwise. Abusing notation, we shall also denote this policy by S . This allows us to reformulate stochastic optimization problem (2) as the *combinatorial optimization problem* of finding a subset $S \in 2^N$ whose associated cost $Z^S(\mathbf{C}^0, \mathbf{C}^1)$ is minimum:

$$Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1) = \min \{ Z^S(\mathbf{C}^0, \mathbf{C}^1) : S \in 2^N \}. \quad (3)$$

Formulation (3) highlights the fundamental *combinatorial* nature of *stochastic* optimization problem (2).

2.2 Does DP provide an efficient solution?

Classical DP theory provides a *theoretically efficient* solution procedure for problem (2), which could lead us to consider it “well solved,” and elementary. We argue in this section that such is far from being the case, in general, from the practical viewpoint of actually designing and implementing an optimal policy in a large-scale model.

In particular, it follows from DP theory that, in spite of the exponential size of the solution space for problem (3), it can be solved in *polynomial time* on the size of state space N . This is a direct consequence of two well-known results: (i) the polynomial size of the standard LP formulation for MDC problem (1), which involves $2n$ variables and constraints, where $n = |N|$ is the cardinality of the state space; and (ii) the polynomial-time solvability of LP.

Yet in large-scale models, where the size of the state space grows exponentially on the size of the model’s defining parameters, such a purely computational approach does *not* yield a *practically efficient* solution procedure (one that runs in polynomial time on the size of the model’s parameters).

2.3 Establishing structural properties on optimal policies

The above discussion motivates the practical interest of identifying conditions on model parameters under which a *restricted* family of policies, which possess a given structural property, is optimal for problem (2), under a suitably restricted range of linear cost coefficients. From the setting above, a family of policies can be characterized by a given family $\mathcal{F} \subseteq 2^N$ of state

subsets, where the *active* action can be taken. Note that sets $S \in \mathcal{F}$ play the role, when interpreted as policies, of *switching surfaces* in state space N , as they indicate when to switch from one action to the other.

Consider now the following problems:

Problem 1. Obtain sufficient conditions on model parameters under which

$$Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1) = \min \{Z^S(\mathbf{C}^0, \mathbf{C}^1) : S \in 2^N\} = \min \{Z^S(\mathbf{C}^0, \mathbf{C}^1) : S \in \mathcal{F}\}, \quad (4)$$

so that there exists an optimal policy within the given family \mathcal{F} .

Problem 2. Design an efficient *combinatorial algorithm* for finding an optimal set/policy $S^{\text{OPT}} \in \mathcal{F}$.

We shall return to problems 1 and 2 above in Section 5. The solutions we shall present there will be obtained by casting the problems into the general polyhedral framework to be developed next, in Sections 3 and 4 below. In our framework, we shall assume that *set system* (N, \mathcal{F}) is a *greedoid* (cf. [12]), which further satisfies $N \in \mathcal{F}$, as this seems to us the appropriate combinatorial model for a switching surface.

3 Extended greedoid polyhedra

We introduce in this section a new type of polyhedron associated with a greedoid, which generalizes the *extended polymatroids* identified in [19] and further investigated in [2], [1]. As we shall show below, such greedoid polyhedra, like extended polymatroids, possess strong structural and algorithmic properties. They shall play a central role in the polyhedral framework for stochastic scheduling problems to be presented in Section 4, in which they provide strong LP relaxations, which, under appropriate conditions, are tight and characterize optimal policies.

Let N be a finite set with cardinality $|N| = n$, and let \mathcal{F} be a collection of subsets of N , i.e., $\mathcal{F} \subseteq 2^N$. Let $b(S)$ be a nonnegative set function defined on $S \in \mathcal{F}$, and let $\mathbb{A} = (A_e^S)_{e \in N, S \in \mathcal{F}}$ be a *weight matrix* that satisfies

$$A_e^S > 0, \quad e \in S, S \in \mathcal{F}. \quad (5)$$

We shall assume that $(N, \mathcal{F}; \mathbb{A})$ is a *weighted greedoid*, as defined next, which further satisfies $N \in \mathcal{F}$.

Definition 1 (Weighted greedoid) *Set system* (N, \mathcal{F}) is a greedoid if the following conditions hold:

- (i) $\emptyset \in \mathcal{F}$;
- (ii) $S \in \mathcal{F}, S \neq \emptyset \implies$ there exists $e \in S$ such that $S \setminus \{e\} \in \mathcal{F}$;
- (iii) $S_1, S_2 \in \mathcal{F}, |S_1| > |S_2| \implies$ there exists $e \in S_1 \setminus S_2$ such that $S_2 \cup \{e\} \in \mathcal{F}$.

If, furthermore, $\mathbb{A} = (A_e^S)_{e \in N, S \in \mathcal{F}}$ is a real weight matrix, we say that $(N, \mathcal{F}; \mathbb{A})$ is a *weighted greedoid*.

Greedoids were introduced in [12], as a characterization of structures in which the greedy algorithm “works”, in an appropriate sense. We define next the notion of (*full*) *ordered string* of a greedoid.

3.1 LP over extended greedoid polyhedra and adaptive-greedy algorithm

Consider the LP problem of minimizing a linear cost objective, with coefficients vector $\mathbf{C} = (C_e)_{e \in N}$, over extended greedoid polytope $\mathcal{P}(\mathbb{A}, b)$:

$$Z^{\text{LP}}(\mathbf{C}) = \min \left\{ \sum_{e \in N} C_e x_e : \mathbf{x} \in \mathcal{P}(\mathbb{A}, b) \right\}. \quad (9)$$

Consider now the following related questions:

1. Given $\mathbf{e} \in \mathcal{E}(\mathcal{F})$, Is extreme point $\mathbf{x}(\mathbf{e})$ optimal for LP (9)?
2. What is the region $\mathcal{C} \subseteq \mathbb{R}^N$ of *feasible cost vectors* \mathbf{C} for which there exists an optimal solution to LP (9) of the form $\mathbf{x}(\mathbf{e})$, for some $\mathbf{e} \in \mathcal{E}(\mathcal{F})$? How to find such an optimal solution?

We develop in this section a common algorithmic answer to questions 1 and 2 above. Our main result will be given in Theorem 2 below, which concerns the *adaptive-greedy algorithm* $\text{AG}(\cdot | \mathbb{A}, N, \mathcal{F})$ described in Figure 1. This algorithm takes as input cost vector \mathbf{C} , and returns an output of the form $(\text{FEASIBLE}, \boldsymbol{\gamma}, \mathbf{e})$. In the output, *FEASIBLE* is a flag that can return the value *TRUE* or *FALSE*. When *FEASIBLE* = *TRUE*, the algorithm further computes a real index vector $\boldsymbol{\gamma} = (\gamma_e)_{e \in N}$, and an ordered string $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$. In short, it will be shown in what follows that the answers to the above questions are:

1. Extreme point $\mathbf{x}(\mathbf{e})$ is optimal if and only if

$$\gamma_{e_n} \leq \dots \leq \gamma_{e_1}.$$

2. Cost vector \mathbf{C} belongs in region \mathcal{C} if and only if *FEASIBLE* = *TRUE*. In such case, the ordered string $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ returned by the algorithm is optimal (i.e., $\mathbf{x}(\mathbf{e})$ is optimal), as it satisfies the optimality condition in point 1 above.

We next state formally the result. A proof for it will be given later in the section.

Theorem 2 (Feasibility/optimality by adaptive-greedy algorithm) *The following results hold:*

(a) $\mathbf{C} \in \mathcal{C}$ if and only if *FEASIBLE* = *TRUE*.

Furthermore, if *FEASIBLE* = *TRUE*, then

(b) $\boldsymbol{\gamma}$ is a univocally determined index vector that characterizes the optimal solutions to LP (9): they are of the form $\mathbf{x}(\mathbf{e})$, where $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$ satisfies the optimality condition

$$\gamma_{e_n} \leq \dots \leq \gamma_{e_1}. \quad (10)$$

Furthermore, the $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ produced by the algorithm satisfies (10).

Algorithm $\text{AG}(\cdot | \mathbb{A}, N, \mathcal{F})$ is an extension of the adaptive-greedy algorithm introduced by Klimov in [11], which was adapted to the solution of LP problems over extended polymatroids in [19], [2], [1]. In particular, Klimov's algorithm corresponds to the special case of algorithm $\text{AG}(\cdot | \mathbb{A}, N, \mathcal{F})$ where $\mathcal{F} = 2^N$, so that, in that case, $\mathcal{C} = \mathbb{R}^N$ and hence *FEASIBLE* always returns the value *TRUE*.

ADAPTIVE-GREEDY ALGORITHM $\text{AG}(\cdot|\mathbb{A}, N, \mathcal{F})$:**Input:** $\mathbf{C} = (C_e)_{e \in N}$ **Output:** $(\text{FEASIBLE}, \gamma, \mathbf{e})$,where $\text{FEASIBLE} \in \{\text{TRUE}, \text{FALSE}\}$, $\gamma = (\gamma_e)_{e \in N}$, and $\mathbf{e} = (e_1, \dots, e_n)$ is a full ordered string of (N, \mathcal{F}) .*Initialization:***let** $\text{FEASIBLE} := \text{TRUE}$ **choose** $e_n \in \arg \min \{C_e/A_e^N : e \in N, N \setminus \{e\} \in \mathcal{F}\}$ **let** $\gamma_{e_n} := \bar{y}^N := C_{e_n}/A_{e_n}^N$ **if** $\bar{y}^N > \min \{C_e/A_e^N : e \in N\}$ **then** **let** $\text{FEASIBLE} := \text{FALSE}$ **end** {if}**let** $S_n = N$ *Loop:***let** $k := 1$ **while** $\text{FEASIBLE} := \text{TRUE}$ **and** $k \leq n$ **do** **let** $S_{n-k} := S_{n-k+1} \setminus \{e_{n-k+1}\}$ **choose** $e_{n-k} \in \arg \min \left\{ \left[C_e - \sum_{j=0}^{k-1} A_e^{S_{n-j}} \bar{y}^{S_{n-j}} \right] / A_e^{S_{n-k}} : e \in S_{n-k}, S_{n-k} \setminus \{e\} \in \mathcal{F} \right\}$ **let** $\bar{y}^{S_{n-k}} := \left[C_{e_{n-k}} - \sum_{j=0}^{k-1} A_{e_{n-k}}^{S_{n-j}} \bar{y}^{S_{n-j}} \right] / A_{e_{n-k}}^{S_{n-k}}$ **let** $\gamma_{e_{n-k}} := \gamma_{e_{n-k+1}} + \bar{y}^{S_{n-k}}$ **if** $\bar{y}^{S_{n-k}} > \min \left\{ \left[C_e - \sum_{j=0}^{k-1} A_e^{S_{n-j}} \bar{y}^{S_{n-j}} \right] / A_e^{S_{n-k}} : e \in S_{n-k} \right\}$ **then** **let** $\text{FEASIBLE} := \text{FALSE}$ **end** {if} **let** $k := k + 1$ **end** {while}Figure 1: Adaptive-greedy algorithm for weighted greedoid $(N, \mathcal{F}; \mathbb{A})$.

In the rest of this section we develop several preliminary results on the structure of optimal solutions for LP (9), which we shall use later for presenting a proof of Theorem 2.

Our analysis of LP (9) is based on consideration of its dual problem. The dual LP, which must have the same optimum value $Z^{\text{LP}}(\mathbf{C})$, is formulated by introducing a dual variable y^S for each $S \in \mathcal{F}$:

$$\begin{aligned}
 Z^{\text{LP}}(\mathbf{C}) &= \max \sum_{S \in \mathcal{F}} b(S) y^S & (11) \\
 &\text{subject to} \\
 &\sum_{S: e \in S \in \mathcal{F}} A_e^S y^S \leq C_e, \quad e \in N \\
 &y^S \geq 0, \quad S \in \mathcal{F} \setminus \{N\} \\
 &y^N \text{ unrestricted.}
 \end{aligned}$$

Given $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$, and variable vector $\mathbf{y} = (y^S)_{S \in \mathcal{F}} \in \mathbb{R}^{\mathcal{F}}$, let us write

$$\mathbf{y}^{\mathbf{e}} = (y^{\{e_1\}}, \dots, y^{\{e_1, \dots, e_n\}}),$$

and let further

$$\mathbf{C}^{\mathbf{e}} = (C_{e_1}, \dots, C_{e_n}).$$

Consider now the triangular system of linear equations

$$A_{e_i}^{\{e_1, \dots, e_i\}} y^{\{e_1, \dots, e_i\}} + \dots + A_{e_i}^{\{e_1, \dots, e_n\}} y^{\{e_1, \dots, e_n\}} = C_{e_i}, \quad 1 \leq i \leq n,$$

or, in the above matrix notation,

$$\mathbf{y}^{\mathbf{e}} \mathbb{A}^{\mathbf{e}} = \mathbf{C}^{\mathbf{e}}, \quad (12)$$

and let us denote its unique solution by $\mathbf{y}^{\mathbf{e}}(\mathbf{e}) = (y^{\{e_1, \dots, e_j\}}(\mathbf{e}))_{1 \leq j \leq n}$. Let us further extend this solution into a vector $\mathbf{y}(\mathbf{e}) = (y^S(\mathbf{e}))_{S \in \mathcal{F}}$ by letting $y^S(\mathbf{e}) = 0$ for all other sets $S \in \mathcal{F}$ that are not of the form $S = \{e_1, \dots, e_j\}$. Note that the definition of the $y^{\{e_1, \dots, e_j\}}(\mathbf{e})$'s implies that, for *any* real vector $\mathbf{x} = (x_e)_{e \in N}$, the linear objective in LP (9) can be reformulated as

$$\sum_{e \in N} C_e x_e = \sum_{S \in \mathcal{F}} y^S(\mathbf{e}) \sum_{e \in S} A_e^S x_e = \sum_{j=1}^n y^{\{e_1, \dots, e_j\}}(\mathbf{e}) \sum_{e \in \{e_1, \dots, e_j\}} A_e^{\{e_1, \dots, e_j\}} x_e. \quad (13)$$

Let \mathbb{T} denote the $n \times n$ lower triangular matrix given by

$$\mathbb{T} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}, \quad (14)$$

and let us define, for each $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$, the real *index* vector $\boldsymbol{\gamma}(\mathbf{e}) = (\gamma_e(\mathbf{e}))_{e \in N}$ given by

$$\gamma_{e_j}(\mathbf{e}) = \sum_{S \in \mathcal{F}} y^S(\mathbf{e}) = \sum_{i=j}^n y^{\{e_1, \dots, e_i\}}(\mathbf{e}), \quad 1 \leq j \leq n,$$

or, in the matrix notation above, letting $\boldsymbol{\gamma}^{\mathbf{e}}(\mathbf{e}) = (\gamma_{e_1}(\mathbf{e}), \dots, \gamma_{e_n}(\mathbf{e}))$,

$$\boldsymbol{\gamma}^{\mathbf{e}}(\mathbf{e}) = \mathbf{y}^{\mathbf{e}}(\mathbf{e}) \mathbb{T} = \mathbf{C}^{\mathbf{e}} (\mathbb{A}^{\mathbf{e}})^{-1} \mathbb{T}. \quad (15)$$

Let us return now to question 1 above: given $\mathbf{e} \in \mathcal{E}(\mathcal{F})$, How can it be tested whether extreme point $\mathbf{x}(\mathbf{e})$ is optimal? The next result gives the required optimality test.

Lemma 3 (Optimality conditions) *The following results hold:*

(a) *if the condition*

$$\mathbf{y}^{\mathbf{e}}(\mathbf{e}) = \mathbf{C}^{\mathbf{e}} (\mathbb{A}^{\mathbf{e}})^{-1} \in \mathbb{R}_+^{n-1} \times \mathbb{R}, \quad (16)$$

or, equivalently,

$$\gamma_{e_n}(\mathbf{e}) \leq \gamma_{e_{n-1}}(\mathbf{e}) \leq \dots \leq \gamma_{e_1}(\mathbf{e}), \quad (17)$$

holds, then $\mathbf{x}(\mathbf{e})$ and $\mathbf{y}(\mathbf{e})$ are an optimal primal-dual solution pair for LP (9) and its dual LP (11);

(b) if condition (16) (or (17)) fails to hold, and extreme point $\mathbf{x}(\mathbf{e})$ is nondegenerate, i.e., $\mathbf{x}(\mathbf{e}) > \mathbf{0}$ componentwise, then $\mathbf{x}(\mathbf{e})$ is not an optimal solution to LP (9).

Proof

(a) It is straightforward to verify that conditions (16) and (17) are equivalent. Suppose $y^{\{e_1, \dots, e_j\}}(\mathbf{e}) \geq 0$, for $1 \leq j \leq n-1$. Then, vector $\mathbf{y}(\mathbf{e})$ is a feasible solution to dual LP (11). Furthermore, vectors $\mathbf{x}(\mathbf{e})$ and $\mathbf{y}(\mathbf{e})$ satisfy LP complementary slackness, as follows from their definitions. Therefore, they are an optimal primal-dual solution pair as stated.

(b) The result follows from standard LP theory, since the standard LP optimality criterion, given by (16), is well known to be necessary when applied to a nondegenerate basic feasible primal solution. \square

Let us consider next question 2 above: For which cost vectors \mathbf{C} does there exist an optimal solution to LP (9) of the form $\mathbf{x}(\mathbf{e})$? The answer is suggested by the optimality conditions in Lemma 3. Let

$$\begin{aligned} \mathcal{C} &= \{ \mathbf{C} \in \mathbb{R}^N : \text{there exists } \mathbf{e} \in \mathcal{E}(\mathcal{F}) \text{ with } \mathbf{C}^{\mathbf{e}} (\mathbb{A}^{\mathbf{e}})^{-1} \in \mathbb{R}_+^{n-1} \times \mathbb{R} \} \\ &= \{ \mathbf{C} \in \mathbb{R}^N : \text{there exists } \mathbf{e} \in \mathcal{E}(\mathcal{F}) \text{ with } \gamma_{e_n}(\mathbf{e}) \leq \gamma_{e_{n-1}}(\mathbf{e}) \leq \dots \leq \gamma_{e_1}(\mathbf{e}) \}. \end{aligned}$$

Definition 4 (Feasible cost region) \mathcal{C} is called the feasible cost region for weighted greedoid $(N, \mathcal{F}; \mathbb{A})$.

The next result shows that region \mathcal{C} provides the required answer to question 2 above.

Lemma 4 *There exists an optimal solution for LP (9) of the form $\mathbf{x}(\mathbf{e})$, for some $\mathbf{e} \in \mathcal{E}(\mathcal{F})$, if and only if $\mathbf{C} \in \mathcal{C}$.*

Proof

The result is a direct consequence of the definition of \mathcal{C} above, and Lemma 3(a). \square

In order to characterize \mathcal{C} more explicitly, let us define, for each $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ as above, the region $\mathcal{C}(\mathbf{e}) \subseteq \mathbb{R}^N$ given by

$$\mathcal{C}(\mathbf{e}) = \left\{ \mathbf{C} \in \mathbb{R}^N : \mathbf{y}^{\mathbf{e}}(\mathbf{e}) = \mathbf{C}^{\mathbf{e}} (\mathbb{A}^{\mathbf{e}})^{-1} \in \mathbb{R}_+^{n-1} \times \mathbb{R} \right\}.$$

Clearly, $\mathcal{C}(\mathbf{e})$ is a polyhedron and, furthermore,

$$\mathcal{C} = \bigcup_{\mathbf{e} \in \mathcal{E}(\mathcal{F})} \mathcal{C}(\mathbf{e}).$$

Therefore, feasible cost region \mathcal{C} is a finite union of polyhedra, though not necessarily a polyhedron itself.

In the next result we establish an important invariance property of dual vector $\mathbf{y}(\mathbf{e})$ and index vector $\gamma(\mathbf{e})$ defined above.

Lemma 5 (Invariance of indices/dual solution) *Dual vector $\mathbf{y}(\mathbf{e})$ and index vector $\gamma(\mathbf{e})$ are invariant under ordered strings $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ that satisfy the optimality conditions in Lemma 3(a).*

Proof

Let $\mathbf{e} = (e_1, \dots, e_n)$ be such an ordered string, and let us write $S_i = \{e_1, \dots, e_i\}$, for $1 \leq i \leq n$ (recall $S_n = N$). Then, by definition of $\mathbf{y}(\mathbf{e}) = (y^S(\mathbf{e}))_{S \in \mathcal{F}}$, the values $y^{S_i}(\mathbf{e})$, for $1 \leq i \leq n$, are characterized as the unique solution of triangular linear system

$$A_{e_i}^{S_i} y^{S_i} + \dots + A_{e_i}^{S_n} y^{S_n} = C_{e_i}, \quad 1 \leq i \leq n. \tag{18}$$

Suppose the optimality conditions in Lemma 3(a) hold, i.e.,

$$y^{S_i} \geq 0, \quad 1 \leq i \leq n-1. \quad (19)$$

Consider the nontrivial case where, among the sets S_1, \dots, S_{n-1} , there is some S_i for which $y^{S_i} > 0$. Label such sets as $S_{i_1} \subset \dots \subset S_{i_K} \subseteq S_n$, for the appropriate $1 \leq K \leq n-1$. We shall establish next that sets S_{i_1}, \dots, S_{i_K} do not depend on \mathbf{e} : they only depend on the satisfaction of the optimality conditions above, and that they further characterize $\mathbf{y}(\mathbf{e})$, which will prove the result.

The definition of sets S_{i_1}, \dots, S_{i_K} , combined with (18), directly yields that the values of $y^{S_{i_1}}, \dots, y^{S_{i_K}}, y^N$ are characterized as the solution of linear system

$$\begin{aligned} A_e^N y^N &= C_e, & e \in N \setminus S_{i_K} \\ A_e^{S_{i_K}} y^{S_{i_K}} + A_e^N y^N &= C_e, & e \in S_{i_K} \setminus S_{i_{K-1}} \\ A_e^{S_{i_k}} y^{S_{i_k}} + \dots + A_e^{S_{i_K}} y^{S_{i_K}} + A_e^N y^N &= C_e, & e \in S_{i_k} \setminus S_{i_{k-1}}, \quad 2 \leq k \leq K \\ A_e^{S_{i_1}} y^{S_{i_1}} + \dots + A_e^{S_{i_K}} y^{S_{i_K}} + A_e^N y^N &= C_e, & e \in S_{i_1}. \end{aligned}$$

This linear system, combined with the assumption that $y^{S_{i_k}} > 0$, for $1 \leq k \leq K$, implies that

$$y^N = \min \left\{ \frac{C_e}{A_e^N} : e \in N \right\},$$

and

$$N \setminus S_{i_K} = \arg \min \left\{ \frac{C_e}{A_e^N} : e \in N \right\},$$

so that S_{i_K} and y^N are univocally determined, independently of the specific choice of $\mathbf{e} \in \mathcal{E}$, as long as it satisfies the required optimality conditions.

Similarly, we have

$$y^{S_{i_K}} = \min \left\{ \frac{C_e - A_e^N y^N}{A_e^{S_{i_K}}} : e \in S_{i_K} \right\}$$

and

$$S_{i_K} \setminus S_{i_{K-1}} = \arg \min \left\{ \frac{C_e - A_e^N y^N}{A_e^{S_{i_K}}} : e \in S_{i_K} \right\},$$

so that $S_{i_{K-1}}$ and $y^{S_{i_K}}$ are also univocally determined.

Arguing inductively, assume now that S_{i_k}, \dots, S_{i_K} , and hence $y^{S_{i_{k+1}}}, \dots, y^{S_{i_K}}, y^N$ are also determined independently of the choice of \mathbf{e} . Then, it follows from the above that

$$y^{S_{i_k}} = \min \left\{ \frac{C_e - A_e^N y^N - A_e^{S_{i_K}} y^{S_{i_K}} - \dots - A_e^{S_{i_{k+1}}} y^{S_{i_{k+1}}}}{A_e^{S_{i_k}}} : e \in S_{i_k} \right\}$$

and

$$S_{i_k} \setminus S_{i_{k-1}} = \arg \min \left\{ \frac{C_e - A_e^N y^N - A_e^{S_{i_K}} y^{S_{i_K}} - \dots - A_e^{S_{i_{k+1}}} y^{S_{i_{k+1}}}}{A_e^{S_{i_k}}} : e \in S_{i_k} \right\},$$

so that also $S_{i_{k-1}}$ and $y^{S_{i_k}}$ are determined independently of \mathbf{e} .

This proves the result for $\mathbf{y}(\mathbf{e})$. Since $\gamma(\mathbf{e})$ is a linear function of $\mathbf{y}(\mathbf{e})$, it must also be invariant in the same sense, which completes the proof. \square

In light of Lemma 5, we shall write in what follows \mathbf{y} and γ instead of $\mathbf{y}(\mathbf{e})$ and $\gamma(\mathbf{e})$, when $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ satisfies the required optimality conditions. Note that \mathbf{y} and γ can be regarded as functions of cost vector \mathbf{C} , defined on domain \mathcal{C} . Given the central role of the indices γ_e in the solution of LP (9), we next define them formally.

Definition 5 (Allocation indices) *Given $\mathbf{C} \in \mathcal{C}$, we say that the corresponding $\gamma = (\gamma_e)_{e \in N}$ is the vector of allocation indices for cost vector \mathbf{C} with respect to weighted greedoid $(N, \mathcal{F}; \mathbb{A})$.*

The next result states a fundamental *indexability property* of LP (9). It says that, for a feasible cost vector $\mathbf{C} \in \mathcal{C}$, the optimal solutions to LP (9) of the form $\mathbf{x}(\mathbf{e})$ as above are fully characterized by the allocation indices in Definition 5. Let $\mathbf{C} \in \mathcal{C}$, and let $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$. Let further $\gamma = (\gamma_e)_{e \in N}$ be the corresponding allocation index vector in Definition 5.

Theorem 6 (Indexability) *Vector $\mathbf{x}(\mathbf{e})$ solves LP (9) optimally if and only if*

$$\gamma_{e_n} \leq \dots \leq \gamma_{e_1}. \quad (20)$$

Furthermore, if (20) holds, then the optimal value of LP (9) is

$$Z^{\text{OPT}}(\mathbf{C}) = \gamma_{e_n} \sum_{e \in N} A_e^N x_e(\mathbf{e}) + \sum_{i=1}^{n-1} (\gamma_{e_i} - \gamma_{e_{i+1}}) \sum_{e \in \{e_1, \dots, e_i\}} A_e^{\{e_1, \dots, e_i\}} x_e(\mathbf{e}). \quad (21)$$

Proof

The first result follows by combining optimality condition (17) in Lemma 3(a) with the invariance result in Lemma 5.

Identity (21) follows by combining (13), the definition of allocation indices, and the result that $\mathbf{x}(\mathbf{e})$ is optimal under (20). \square

3.2 Analysis of adaptive-greedy algorithm

We now return to algorithm $\text{AG}(\cdot | \mathbb{A}, N, \mathcal{F})$, analyzing it in the light of the previous results. The algorithm has two related functions: (1) to test whether its input \mathbf{C} is feasible ($\mathbf{C} \in \mathcal{C}$); and, (2) if so, to construct the vector $\gamma = (\gamma_e)_{e \in N}$ of allocation indices for LP (9) (see Definition 5), as well as an ordered string $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$ that satisfies optimality condition (20) in Theorem 6, and therefore yields an optimal solution $\mathbf{x}(\mathbf{e})$ to LP (9).

Each step of the algorithm consists of two parts with different purposes: (1) an *element selection* part, which selects (in an *adaptive-greedy* fashion, hence the algorithm's name) the next element $e \in N$ to be appended at the right end of the current partially constructed ordered string; and (2) a *feasibility test*, which checks whether current information implies that it is not possible to construct an ordered string \mathbf{e} satisfying (16), i.e., whether $\mathbf{C} \in \mathcal{C}$. The algorithm proceeds to completion, in which case it takes $|N| = n$ steps, only in the case that each feasibility test is positive, and it exits prematurely otherwise.

Regarding the complexity of the algorithm, let us assume that, for each $S \in \mathcal{F}$ and $e \in S$, the value A_e^S is returned by an *oracle*. Let us further assume that another oracle provides a *membership test* for determining whether $S \in \mathcal{F}$, for each $S \in 2^N$. Then, it is immediate from the algorithm's description that the number of calls it makes to these oracles is $O(n^2)$.

Furthermore, the algorithm performs $O(n^3)$ multiplications and $O(n^2)$ pairwise comparisons. Therefore, in the case where both oracles run in polynomial time on the model's defining data, $\text{AG}(\cdot|\mathbb{A}, N, \mathcal{F})$ is a polynomial-time algorithm. adaptive-greedy algorithm $\text{AG}(\cdot|\mathbb{A}, N, \mathcal{F})$.

We next present the required proof for the validity of the adaptive-greedy algorithm. Let $(\text{FEASIBLE}, \gamma, \mathbf{e})$ be an output of algorithm $\text{AG}(\cdot|\mathbb{A}, N, \mathcal{F})$ corresponding to a given input cost vector \mathbf{C} .

Proof of Theorem 2

(a) It is immediate from the algorithm's description in Figure 1 that it constructs an ordered string \mathbf{e} of greedoid (N, \mathcal{F}) spanning N for which $\mathbf{y}(\mathbf{e})$ satisfies optimality criterion (16), so that $\text{FEASIBLE} = \text{TRUE}$, if and only if such an ordered string exists, i.e., if and only if $\mathbf{C} \in \mathcal{C}$.

(b) If $\text{FEASIBLE} = \text{TRUE}$, let \mathbf{e} be an ordered string such that $\mathbf{y}(\mathbf{e})$ satisfies optimality condition (16). Then, by the invariance result in Lemma 5, and Definition 5, it follows from the algorithm's description that it constructs the vector γ of allocation indices for LP (9), and that this satisfies the optimality condition in Theorem 6(b). Hence $\mathbf{x}(\mathbf{e})$ is optimal, which completes the proof. \square

3.3 Feasible cost region and index decomposition

Under additional conditions, both the feasible cost region \mathcal{C} and the indices γ_e above possess strong *decomposition properties*, which dramatically simplify their computation. Suppose weighted greedoid $(N, \mathcal{F}; \mathbb{A})$ can be *decomposed* into $K \geq 2$ weighted greedoids $(N_k, \mathcal{F}_k; \mathbb{A}^k)$, where $\mathbb{A}^k = (A_e^S)_{e \in N_k, S \in \mathcal{F}^k}$, for $1 \leq k \leq K$, satisfying the conditions given next.

Assumption 1 *The following conditions hold:*

- (i) sets N_1, \dots, N_K form a partition of N ;
- (ii) (N_k, \mathcal{F}_k) is a greedoid with $N_k \in \mathcal{F}_k$;
- (iii) $\mathcal{F} = \{S_1 \cup \dots \cup S_K : S_k \in \mathcal{F}_k, 1 \leq k \leq K\}$;
- (iv) $A_e^S = A_e^{S \cap N_k}$, for $S \in \mathcal{F}$, $e \in S \cap N_k$.

Note that it follows from Assumption 1 that $\mathcal{F}_k = \{S \cap N_k : S \in \mathcal{F}\}$. Let us further denote by $\mathcal{E}(\mathcal{F}_k)$ the set of all full ordered strings of greedoid (N_k, \mathcal{F}_k) , and let \mathbb{T}^k be a matrix with the same structure as \mathbb{T} , given by (14), having dimension $n(k) \times n(k)$, where $n(k) = |N_k|$, for $1 \leq k \leq K$. Our next result will play a key role in our proof of the decomposition results in Theorem 8 below.

Let $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$. It should be clear at this point that Assumption 1 implies that \mathbf{e} induces a corresponding full ordered string $\mathbf{e}^k \in \mathcal{E}(\mathcal{F}_k)$ for each k . Let us write $\mathbf{e}^k = (e_{i(k,1)}, \dots, e_{i(k,n(k))})$. For example, if $\mathbf{e} = (e_1, e_2, e_3)$, $N_1 = \{e_2\}$ and $N_2 = \{e_1, e_3\}$, then $\mathbf{e}^1 = (e_2)$ and $\mathbf{e}^2 = (e_1, e_3)$. Let us further write \mathbb{A}^e and $\mathbb{A}^{k, \mathbf{e}^k}$ as in (6), and let $\gamma^e(\mathbf{e}) \in \mathbb{R}^n$ and $\gamma^{\mathbf{e}^k}(\mathbf{e}^k) \in \mathbb{R}^{n(k)}$ be the appropriate index vectors as defined in (15). To simplify the exposition in what follows, we shall consider (without loss of generality) that

$$\mathbf{e} = (\mathbf{e}^1, \dots, \mathbf{e}^K).$$

Lemma 7 *Under Assumption 1, the following results hold:*

(a)

$$\mathbb{T}^{-1}\mathbb{A}^{\mathbf{e}} = \begin{pmatrix} (\mathbb{T}^1)^{-1}\mathbb{A}^{1,\mathbf{e}^1} & & & \\ & (\mathbb{T}^2)^{-1}\mathbb{A}^{2,\mathbf{e}^2} & & \\ & & \ddots & \\ & & & (\mathbb{T}^K)^{-1}\mathbb{A}^{K,\mathbf{e}^K} \end{pmatrix};$$

(b)

$$\gamma^{\mathbf{e}}(\mathbf{e}) = (\gamma^{\mathbf{e}^1}(\mathbf{e}^1) \leq \dots \leq \gamma^{\mathbf{e}^K}(\mathbf{e}^K)).$$

Proof(a) From the definition of \mathbb{T} in (14), it follows that

$$\mathbb{T}^{-1} = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix},$$

hence,

$$\mathbb{T}^{-1}\mathbb{A}^{\mathbf{e}} = \begin{pmatrix} A_{e_1}^{\{e_1\}} & 0 & \dots & 0 \\ A_{e_1}^{\{e_1,e_2\}} - A_{e_1}^{\{e_1\}} & A_{e_2}^{\{e_1,e_2\}} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_{e_1}^{\{e_1,\dots,e_n\}} - A_{e_1}^{\{e_1,\dots,e_{n-1}\}} & \dots & A_{e_{n-1}}^{\{e_1,\dots,e_n\}} - A_{e_{n-1}}^{\{e_1,\dots,e_{n-1}\}} & A_{e_n}^{\{e_1,\dots,e_n\}} \end{pmatrix}$$

and similarly, for $1 \leq k \leq K$, $(\mathbb{T}^k)^{-1}\mathbb{A}^{k,\mathbf{e}^k}$ is given by

$$\begin{pmatrix} A_{e_{i(k,1)}}^{\{e_{i(k,1)}\}} & 0 & \dots & 0 \\ A_{e_{i(k,1)}}^{\{e_{i(k,1)},e_{i(k,2)}\}} - A_{e_{i(k,1)}}^{\{e_{i(k,1)}\}} & A_{e_{i(k,2)}}^{\{e_{i(k,1)},e_{i(k,2)}\}} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_{e_{i(k,1)}}^{N_k} - A_{e_{i(k,1)}}^{N_k \setminus \{e_{i(k,n)}\}} & \dots & A_{e_{i(k,n(k)-1)}}^{N_k} - A_{e_{i(k,n(k)-1)}}^{N_k \setminus \{e_{i(k,n)}\}} & A_{e_{i(k,n(k)-1)}}^{N_k} \end{pmatrix}.$$

Notice now that, if $e_i \in N_k$, $e_{j+1} \in N \setminus N_k$ and $i \leq j$, then Assumption 1 implies

$$A_{e_i}^{\{e_1,\dots,e_{j+1}\}} = A_{e_i}^{\{e_1,\dots,e_{j+1}\} \cap N_k} = A_{e_i}^{\{e_1,\dots,e_j\} \cap N_k} = A_{e_i}^{\{e_1,\dots,e_j\}},$$

and, therefore,

$$A_{e_i}^{\{e_1,\dots,e_{j+1}\}} - A_{e_i}^{\{e_1,\dots,e_j\}} = 0. \quad (22)$$

The identities in (22), combined with the expressions for $\mathbb{T}^{-1}\mathbb{A}^{\mathbf{e}}$ and $(\mathbb{T}^k)^{-1}\mathbb{A}^{k,\mathbf{e}^k}$ above prove part (a).(b) This part follows directly by combining part (a) with the definition of the index vectors $\gamma^{\mathbf{e}}(\mathbf{e})$, $(\gamma^{\mathbf{e}^k}(\mathbf{e}^k))$ in (15). \square Let $\mathcal{C} \subseteq \mathbb{R}^N$ be the feasible cost region for weighted greedoid $(N, \mathcal{F}; \mathbb{A})$, and let $\mathcal{C}^k \subseteq \mathbb{R}^{N_k}$ denote the feasible cost region for weighted greedoid $(N_k, \mathcal{F}_k; \mathbb{A}^k)$, for $1 \leq k \leq K$. Given a

cost vector $\mathbf{C} = (C_e)_{e \in N}$, we further write $\mathbf{C}^k = (C_e)_{e \in N_k}$, for each k . If $\mathbf{C} \in \mathcal{C}$, we shall denote by $\gamma = (\gamma_e)_{e \in N}$ the corresponding index vector. Furthermore, if $\mathbf{C}^k \in \mathcal{C}^k$, we shall denote by $\gamma^k = (\gamma_e^k)_{e \in N_k}$ the allocation index vector for \mathbf{C}^k with respect to weighted greedoid $(N_k, \mathcal{F}_k; \mathbb{A}^k)$, for $1 \leq k \leq K$. See Definition 5.

Theorem 8 (Feasible cost region/index decomposition) *Under Assumption 1 above, the following results hold:*

(a) *Feasible cost region \mathcal{C} is decomposed as*

$$\mathcal{C} = \mathcal{C}^1 \times \dots \times \mathcal{C}^K;$$

(b) *if $\mathbf{C} \in \mathcal{C}$, then*

$$\gamma_e = \gamma_e^k, \quad \text{for } 1 \leq k \leq K. \quad (23)$$

Proof

Both results follow directly by combining Lemma 7(b), Definition 5 (allocation indices) and Definition 4 (feasible cost region). \square

Theorem 8 allows to simplify considerably the computational burden required for determining whether a given cost vector \mathbf{C} is feasible, and, if so, for computing the corresponding allocation indices. Recall that, by Theorem 2, such information is given by the output produced by adaptive-greedy algorithm $\text{AG}(\cdot|N, \mathcal{F}; \mathbb{A})$ when fed with input \mathbf{C} . Similarly, the determination of whether cost vector $\mathbf{C}^k = (C_e)_{e \in N_k}$ is feasible, and the computation of the corresponding allocation indices γ_e^k in the positive case, is accomplished by algorithm $\text{AG}(\cdot|N_k, \mathcal{F}_k; \mathbb{A}^k)$ when fed with input $\mathbf{C}^k = (C_e)_{e \in N_k}$.

Therefore, Theorem 8 says that, under the required assumptions, the determination of whether vector \mathbf{C} is feasible, and the corresponding index computation, can be carried out by running independently (e.g., *in parallel*) adaptive-greedy algorithm $\text{AG}(\cdot|N_k, \mathcal{F}_k; \mathbb{A}^k)$ on input $\mathbf{C}^k = (C_e)_{e \in N_k}$, for $1 \leq k \leq K$, which dramatically reduces the computation time.

4 A polyhedral framework for partially indexable scheduling problems

In this section we develop a polyhedral framework, based on the properties of the extended greedoid polyhedra discussed in Section 3, for establishing a wide range of structural properties on optimal solutions for stochastic scheduling problems. This framework extends that given in [1]. Consider a general dynamic and stochastic scheduling model involving a set N of job classes with finite cardinality $|N| = n$. Service resources are to be allocated to jobs on the basis of a *scheduling policy* u , belonging in a space \mathcal{U} of *admissible* policies. The performance of a policy $u \in \mathcal{U}$ over job class $e \in N$ is given by a *performance measure* $x_e(u) \geq 0$ (we write $\mathbf{x}(u) = (x_e(u))_{e \in N}$). We further assume the system admits a consistent notion of *priority* between classes: to each (arbitrary) ordered string of all the elements of N , $\mathbf{e} = (e_1, \dots, e_n)$, we can associate a corresponding *e-policy* (which we shall also denote by \mathbf{e} when there is no confusion), which assigns higher priority to class e_i over class e_j if $i < j$, so that class e_1 has top priority. We further say that a policy gives priority to *S-jobs* if it gives priority to any job class $e \in S$ over any job class $e' \in S^c = N \setminus S$, for $S \in 2^N$.

We shall be concerned with priority policies having a certain special structure, given in the form of a *greedoid* (N, \mathcal{F}) (see Definition 1) having $N \in \mathcal{F}$. Let $\mathbf{e} = (e_1, \dots, e_n)$ be an arbitrary full ordered string of N . Recall that $\mathcal{E}(\mathcal{F})$ denotes the set of *full ordered strings of greedoid* (N, \mathcal{F}) (see Definition 2).

Definition 6 (\mathcal{F} -policy) We say that the \mathbf{e} -policy is an \mathcal{F} -policy if $\mathbf{e} \in \mathcal{E}(\mathcal{F})$.

In light of Definition 6 and the above discussion, the notation $\mathbf{e} \in \mathcal{E}(\mathcal{F})$ will be interpreted in what follows in two possible ways, which will be clear by the context: (1) \mathbf{e} is a full ordered string of greedoid (N, \mathcal{F}) ; or (2) the \mathbf{e} -policy is an \mathcal{F} -policy.

Consider, for a given *cost vector* $\mathbf{C} = (C_e)_{e \in N} \in \mathbb{R}^N$, the *optimal scheduling problem*

$$Z^{\text{OPT}}(\mathbf{C}) = \inf \left\{ \sum_{e \in N} C_e x_e(u) : u \in \mathcal{U} \right\}, \quad (24)$$

which involves finding an optimal policy $u^{\text{OPT}} \in \mathcal{U}$ that solves (24), and computing the optimum value $Z^{\text{OPT}}(\mathbf{C})$. Consider the following questions:

1. How could it be established that \mathcal{F} -policies are optimal for problem (24) among admissible policies, i.e., that

$$Z^{\text{OPT}}(\mathbf{C}) = \min \left\{ \sum_{e \in N} C_e x_e(\mathbf{e}) : \mathbf{e} \in \mathcal{E}(\mathcal{F}) \right\} ?$$

2. When such is the case, How can an optimal \mathcal{F} -policy $\mathbf{e}^{\text{OPT}} \in \mathcal{E}(\mathcal{F})$ be found?

We shall address problems 1 and 2 above through the *achievable region approach*, discussed in the Introduction (cf. [6]). The first step of this approach involves reformulating problem (24) as a mathematical (preferably linear) programming problem. For this purpose we consider the system's *achievable region* \mathcal{X} , spanned by performance vector $\mathbf{x}(u)$ under all admissible policies $u \in \mathcal{U}$, i.e.,

$$\mathcal{X} = \{ \mathbf{x} \in \mathbb{R}^N : \mathbf{x} = \mathbf{x}(u), u \in \mathcal{U} \}. \quad (25)$$

This allows us to reformulate *stochastic* scheduling problem (24) as the *deterministic* mathematical programming problem

$$Z^{\text{OPT}}(\mathbf{C}) = \inf \left\{ \sum_{e \in N} C_e x_e : \mathbf{x} \in \mathcal{X} \right\}, \quad (26)$$

defined on variable vector $\mathbf{x} \in \mathbb{R}^N$.

4.1 Partial conservation laws and LP relaxation

In the framework presented in ([1]), a *full* polyhedral characterization of \mathcal{X} was obtained by assuming that performance vector $\mathbf{x}(u)$ satisfies a set of *generalized conservation laws*. We extend those results by presenting next a set of *partial conservation laws* that imply a *partial* polyhedral characterization of \mathcal{X} . This will provide an *LP relaxation* of problem (26) (and hence of (24)), which is *tight* for some objective functions. The concept of partial conservation laws was recently introduced by the author in [13], in a less developed setting that did not involve the greedoid structure.

Let $\mathbb{A} = (A_e^S)_{e \in N, S \in \mathcal{F}}$ be a *weight* matrix satisfying (5).

Definition 7 (Partial conservation laws (PCL)) Performance vector $\mathbf{x}(u)$ satisfies partial conservation laws with respect to greedoid (N, \mathcal{F}) and weight matrix \mathbb{A} if, letting

$$b(S) = \inf \left\{ \sum_{e \in S} A_e^S x_e(u) : u \in \mathcal{U} \right\}, \quad \text{for } S \in \mathcal{F},$$

the following identities hold: for each $S \in \mathcal{F}$,

$$\sum_{e \in S} A_e^S x_e(\mathbf{e}) = b(S), \quad \text{under any } \mathcal{F}\text{-policy } \mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F}) \text{ with } S = \{e_1, \dots, e_{|S|}\};$$
(27)

and, for $S = N$,

$$\sum_{e \in N} A_e^N x_e(u) = b(N), \quad \text{under any } \mathcal{F}\text{-policy } \mathbf{e} \in \mathcal{E}(\mathcal{F}).$$
(28)

Note that the generalized conservation laws in [1] correspond to the special case where $\mathcal{F} = 2^N$. Assume in what follows that performance vector $\mathbf{x}(u)$ satisfies the PCL in Definition 7. Let $\mathcal{P}(\mathbb{A}, b)$ be the polyhedron defined in (8).

Theorem 9 (Achievable region: partial characterization) $\mathcal{P}(\mathbb{A}, b)$ is an extended greedoid polytope, which contains the achievable region \mathcal{X} . Furthermore, the performance vectors $\mathbf{x}(\mathbf{e})$ achieved by \mathcal{F} -policies \mathbf{e} are vertices of $\mathcal{P}(\mathbb{A}, b)$. Furthermore, in such case the optimum performance objective can be expressed as

$$Z^{\text{OPT}}(\mathbf{C}) = \gamma_{e_n} \sum_{e \in N} A_e^N x_e(\mathbf{e}) + \sum_{i=1}^{n-1} (\gamma_{e_i} - \gamma_{e_{i+1}}) \sum_{e \in \{e_1, \dots, e_i\}} A_e^{\{e_1, \dots, e_i\}} x_e(\mathbf{e}).$$
(29)

Proof

Let $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$. Then, it follows immediately from the definition of PCL above that (1) the vector $\mathbf{x}(\mathbf{e}) = (x_e(\mathbf{e}))_{e \in N}$ defined by the solution of linear equation system (7) is precisely the performance vector of the \mathbf{e} -policy, and $\mathbf{x}(\mathbf{e}) \in \mathcal{P}(\mathbb{A}, b)$; and (2) the \mathbf{e} -policy is an \mathcal{F} -policy. Therefore, $\mathcal{P}(\mathbb{A}, b)$ is an extended greedoid polytope. Furthermore, it follows from Lemma 1 that $\mathbf{x}(\mathbf{e})$ is an extreme point of $\mathcal{P}(\mathbb{A}, b)$, which completes the proof. \square

4.2 Partial indexability under PCL

If performance vector $\mathbf{x}(u)$ satisfies PCL as above, Theorem 9 ensures that LP (9) is an *LP relaxation* of scheduling problem (24), in that $Z^{\text{LP}}(\mathbf{C}) \leq Z^{\text{OPT}}(\mathbf{C})$. Furthermore, since Theorem 9 also guarantees that the performance vectors achieved by \mathcal{F} -policies are vertices of $\mathcal{P}(\mathbb{A}, b)$, it follows that such policies will be optimal for (24) under appropriately restricted cost vectors \mathbf{C} : specifically, for $\mathbf{C} \in \mathcal{C}$, where \mathcal{C} is the *feasible cost region* defined in Section 3.

We thus see that the strong structural properties of extended greedoid polytopes translate directly into corresponding structural properties on the optimal policies for problem (24).

As a direct consequence of the indexability property for extended greedoid polytopes given in Theorem 6, we obtain the following result on the structure of optimal scheduling policies for problem (24). Let $(\text{FEASIBLE}, \boldsymbol{\gamma}, \mathbf{e})$ be an output of adaptive-greedy algorithm $\text{AG}(\cdot | \mathbb{A}, N, \mathcal{F})$ corresponding to input $\mathbf{C} = (C_e)_{e \in N}$. Assume that $\text{FEASIBLE} = \text{TRUE}$, so that $\mathbf{C} \in \mathcal{C}$ and $\boldsymbol{\gamma} = (\gamma_e)_{e \in N}$ is the vector of allocation indices in Definition 5. See Theorem 2.

Theorem 10 (Partial indexability) Any \mathcal{F} -priority policy $\mathbf{e} = (e_1, \dots, e_n) \in \mathcal{E}(\mathcal{F})$ that assigns higher priorities to classes with larger indices γ_e , i.e., which satisfies

$$\gamma_{e_n} \leq \dots \leq \gamma_{e_1},$$

is optimal for scheduling problem (24).

4.3 Project scheduling: an index decomposition property

We consider in this section the important case where scheduling problem (24) corresponds to a *project scheduling* model: there is collection of K *projects*, where project k evolves through a finite *state space* N_k , for $1 \leq k \leq K$; projects must be engaged dynamically according to an admissible *project scheduling policy* $u \in \mathcal{U}$. Note that this problem can be cast in the job scheduling framework above by (1) assuming that the sets N_k are *disjoint*; and (2) identifying each project state with a corresponding job class. Hence, engaging a project in state e corresponds to working on a class e job. The full class set is thus $N = \cup_{k=1}^K N_k$. Note further that the only relevant priorities in a project scheduling model are those between classes/states corresponding to *distinct* projects: this follows since at each time a project is in a given state, so that two job classes corresponding to the same project never compete for service resources. We further assume that the weighted greedoid $(N, \mathcal{F}; \mathbb{A})$ above, which defines the family of \mathcal{F} -policies, is *decomposed* into K corresponding weighted greedoids $(N_k, \mathcal{F}_k; \mathbb{A}^k)$, where

$$\mathcal{F}_k = \{S \cap N_k : S \in \mathcal{F}\}$$

and $\mathbb{A}^k = (A_e^S)_{e \in N_k, S \in \mathcal{F}_k}$, for $1 \leq k \leq K$, which satisfy Assumption 1.

The *index decomposition* property of extended greedoid polyhedra (Theorem 8), combined with Theorem 10, gives directly the next result. Suppose that performance vector $\mathbf{x}(u)$ satisfies the PCL in Definition 7. Assume that the greedoids (N_k, \mathcal{F}_k) , for $1 \leq k \leq K$, satisfy the conditions for Theorem 8. Let us define, for each $1 \leq k \leq K$, the submatrix \mathbb{A}^k of \mathbb{A} by $\mathbb{A}^k = (A_e^S)_{e \in N_k, S \in \mathcal{F}_k}$. Let $(FEASIBLE^k, \gamma^k, \mathbf{e}^k)$ be an output produced by adaptive-greedy algorithm $AG(\cdot | \mathbb{A}^k, N_k, \mathcal{F}_k)$ when fed with input $\mathbf{C}^k = (C_e)_{e \in N_k}$, where $\gamma^k = (\gamma_e^k)_{e \in N_k}$, and assume that $FEASIBLE^k = TRUE$ for each k . Note that the γ_e^k , for $e \in N_k$, can be considered, in this setting, the *allocation indices for the states of project k*.

Theorem 11 (Index decomposition for project scheduling) Any \mathcal{F} -policy that assigns higher priorities to projects whose current states have larger indices is optimal.

5 Partial indexability for restless bandits via PCL

In this section we address the problem of designing an efficient procedure for solving the single restless bandit model discussed in Section 2, by establishing and exploiting appropriate structural properties on optimal policies.

We recall that the *single restless bandit problem* of interest concerns a *single project* modeled as a discrete-time Markov decision chain (MDC), having finite state space N with $|N| = n$, transition probability matrices $\mathbb{P}^a = (p_{ee'}^a)_{e, e' \in N}$, and immediate cost vectors $\mathbf{C}^a = (C_e^a)_{e \in N}$, corresponding to the active ($a = 1$) and passive ($a = 0$) actions, which are available in each state. The decision of which action to take in each state is made through a policy u , chosen from the space \mathcal{U} of nonanticipative policies. Costs are discounted in time by factor $0 < \beta < 1$. The initial state probabilities are given by vector $\mathbf{p} = (p_e)_{e \in N}$, with p_e

being the 0/1 indicator of the initial project state being e . Let us denote by $Z^u(\mathbf{C}^0, \mathbf{C}^1)$ the corresponding total expected discounted cost achieved by policy $u \in \mathcal{U}$, as given in (1). The single restless bandit problem involves finding a policy $u^{\text{OPT}} \in \mathcal{U}$ that achieves the minimum value $Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1)$ of performance objective $Z^u(\mathbf{C}^0, \mathbf{C}^1)$. See (2). As discussed in Section 2, such problem can be reduced to that of finding a state space subset $S \in 2^N$ whose corresponding S -active policy (which takes the active action over S and the passive action over $S^c = N \setminus S$) is optimal among such stationary and deterministic policies. See (3). We shall be concerned with (i) identifying conditions on model parameters under which there exists an optimal policy with a given special structure, characterized by a greedoid (N, \mathcal{F}) having $N \in \mathcal{F}$; see (4); and (ii) obtain efficiently an optimal policy, having the required structure. Such family of \mathcal{F} -policies (cf. Definition 6) consists of all the S -active policies corresponding to sets $S \in \mathcal{F}$. We shall address these problems in the PCL framework developed in Section 4.

For this purpose, we first introduce a *performance measure* $x_e^a(u)$ representing the total expected discounted time that action $a \in \{0, 1\}$ is taken in state $e \in N$ under policy $u \in \mathcal{U}$, i.e., using the notation in (1),

$$x_e^a(u) = E_u \left[\sum_{t=0}^{\infty} 1_{\{X(t)=e, a(t)=a\}} \beta^t \right], \quad (30)$$

where $1_{\{\cdot\}}$ denotes the indicator of the corresponding event. In this way, we can formulate the single restless bandit problem above as

$$Z^{\text{OPT}}(\mathbf{C}^0, \mathbf{C}^1) = \min_{u \in \mathcal{U}} \sum_{e \in N} C_e^0 x_e^0(u) + \sum_{e \in N} C_e^1 x_e^1(u). \quad (31)$$

We next present an extension of an idea introduced by Whittle in [21], which will play a key role in our subsequent PCL analysis: Let us attach to each project state $e \in N$ a parameter $0 < \theta_e \leq 1$, which we shall interpret as a *success probability under the active action in state e* . We write $\boldsymbol{\theta} = (\theta_e)_{e \in N}$. Let us further define an associated parametric problem family as follows: for each $\gamma \in \mathbb{R}$, we consider a related problem where each success is *penalized* with an amount γ . This γ -penalty problem is thus formulated as

$$Z^{\text{OPT}}(\gamma; \mathbf{C}^0, \mathbf{C}^1) = \min_{u \in \mathcal{U}} \sum_{e \in N} C_e^0 x_e^0(u) + \sum_{e \in N} (C_e^1 + \gamma \theta_e) x_e^1(u), \quad \gamma \in \mathbb{R}. \quad (32)$$

Consider the following structural property associated with the family of γ -penalty problems:

Definition 8 ($\boldsymbol{\theta}$ -indexability) *We say that problem (31) is $\boldsymbol{\theta}$ -indexable if the set of states where it is optimal to take the passive action in γ -penalty problem (32) increases monotonically from the empty set to the full state space as the success penalty γ increases from $-\infty$ to $+\infty$.*

We note that Whittle's definition of *indexability* in [21] corresponds to the special case where $\theta_e = 1$, for $e \in N$, in the above setting. Note further that, if problem (31) is $\boldsymbol{\theta}$ -indexable, it follows from Definition 8 that there exist indices γ_e , for $e \in N$, which characterize the optimal solution of the γ -penalty problem for each value of $\gamma \in \mathbb{R}$: take the active action in states $e \in N$ where $\gamma < \gamma_e$, and the passive action otherwise. Note that γ_e is the unique breakpoint for parameter γ where both the active and the passive actions are

optimal in state e . Equivalently, the index γ_e is the smallest value for success penalty γ that would make the active action suboptimal in state e . Note that the possibility of choosing parameters $\theta_e < 1$ allows for index definitions different from those introduced by Whittle, which are obtained by letting each $\theta_e = 1$.

In what follows, we shall deploy the PCL framework to obtain conditions on model parameters under which (i) problem (31) is θ -indexable; and, furthermore, (ii) there exists an optimal \mathcal{F} -policy, characterized by the resulting indices.

5.1 Standard LP formulation, and problem normalization

We review next the standard LP formulation from MDC theory (cf. [17]) for γ -penalty problem (32), which results from taking the dual of an LP formulation of the dynamic programming equations, and show that it implies we can focus on a *normalized problem* where $\mathbf{C}^1 = \mathbf{0}$. We shall further use this LP formulation later on to obtain the required PCL. This standard LP can be written, using vector notation, as

$$Z^{\text{OPT}}(\gamma; \mathbf{C}^0, \mathbf{C}^1) = \min \mathbf{x}^0 \mathbf{C}^0 + \mathbf{x}^1 (\mathbf{C}^1 + \gamma \boldsymbol{\theta}) \quad (33)$$

subject to

$$\mathbf{x}^0 (\mathbb{I} - \beta \mathbb{P}^0) + \mathbf{x}^1 (\mathbb{I} - \beta \mathbb{P}^1) = \mathbf{p} \quad (34)$$

$$\mathbf{x}^0, \mathbf{x}^1 \geq \mathbf{0},$$

where $\mathbf{x}^a = (x_e^a)_{e \in N}$, for $a \in \{0, 1\}$, and \mathbf{p} are taken to be row vectors. Note further that the constraints (34) imply that

$$\mathbf{x}^1 = \mathbf{p} (\mathbb{I} - \beta \mathbb{P}^1)^{-1} - \mathbf{x}^0 (\mathbb{I} - \beta \mathbb{P}^0) (\mathbb{I} - \beta \mathbb{P}^1)^{-1}.$$

A direct consequence of this observation and the above LP is the identity

$$Z^{\text{OPT}}(\gamma; \mathbf{C}^0, \mathbf{C}^1) = \mathbf{p} (\mathbb{I} - \beta \mathbb{P}^1)^{-1} \mathbf{C}^1 + Z^{\text{OPT}}(\gamma; \mathbf{C}^0 - (\mathbb{I} - \beta \mathbb{P}^0) (\mathbb{I} - \beta \mathbb{P}^1)^{-1} \mathbf{C}^1, \mathbf{0}). \quad (35)$$

In light of (35), we shall focus in what follows on the *normalized case* where all active costs are zero, i.e., $\mathbf{C}^1 = \mathbf{0}$, without loss of generality.

5.2 PCL weight matrix definition

We show in this section how to define and construct an appropriate matrix of weights A_e^S for applying the PCL framework. We shall find it convenient to use the following additional notation: given a vector $\mathbf{v} = (v_e)_{e \in N}$, a matrix $\mathbb{B} = (b_{ee'})_{e, e' \in N}$, and subsets $S, T \subseteq N$, we shall write $\mathbf{v}_S = (v_e)_{e \in S}$ and $\mathbb{B}_{ST} = (b_{ee'})_{e \in S, e' \in T}$. We shall further write $S^c = N \setminus S$, for $S \in 2^N$.

We next define certain project parameters, which will play a key role in our results. For each state $e \in N$ and state subset $S \in 2^N$, let us denote by V_e^S the *total expected discounted number of successes under the S -active policy, conditional on the initial state being e* . Note that the V_e^S 's are determined as the unique solution to the linear equation system

$$V_e^S = \theta_e 1\{e \in S\} + 1\{e \in S^c\} \beta \sum_{e' \in N} p_{ee'}^0 V_{e'}^S + 1\{e \in S\} \beta \sum_{e' \in N} p_{ee'}^1 V_{e'}^S, \quad e \in N.$$

It will be convenient to rewrite this system, using the matrix notation above, as

$$\mathbf{V}_S^S = \boldsymbol{\theta}_S + \beta \mathbb{P}_{SS}^1 \mathbf{V}_S^S + \beta \mathbb{P}_{SS^c}^1 \mathbf{V}_{S^c}^S \quad (36)$$

$$\mathbf{V}_{S^c}^S = \beta \mathbb{P}_{S^c S}^0 \mathbf{V}_S^S + \beta \mathbb{P}_{S^c S^c}^0 \mathbf{V}_{S^c}^S, \quad (37)$$

i.e., $\boldsymbol{\theta}_S = (\theta_e)_{e \in S}$, $\mathbf{V}_S^S = (V_e^S)_{e \in S}$ and $\mathbf{V}_{S^c}^S = (V_e^S)_{e \in S^c}$. We shall further write $\mathbf{V}^S = (V_e^S)_{e \in N}$. We next use the V_e^S 's as building blocks to define *weight matrix* $\mathbb{A} = (A_e^S)_{e \in N, S \subseteq N}$, by

$$A_e^S = \theta_e + \beta \sum_{e' \in N} (p_{ee'}^1 - p_{ee'}^0) V_{e'}^S. \quad (38)$$

Note that A_e^S represents the total expected discounted *differential* number of successes that results from taking the active action first, relative to taking instead the passive action, and then proceeding with the S -active policy, conditional on the initial state being e . It follows from (36)–(38) that we can write (38) in matrix notation as

$$\begin{aligned} \mathbf{A}_S^S &= \mathbf{V}_S^S - \beta \mathbb{P}_{SN}^0 \mathbf{V}^S \\ \mathbf{A}_{S^c}^S &= \boldsymbol{\theta}_{S^c} + \beta \mathbb{P}_{S^c N}^1 \mathbf{V}^S - \mathbf{V}_{S^c}^S. \end{aligned} \quad (39)$$

5.3 Reformulation as two-project scheduling problem

In order to cast γ -penalty problem (32) into the PCL framework of Section 4, we shall reformulate it as an equivalent project scheduling problem. See Section 4.3. This involves the dynamic scheduling of *two* projects on a single server: to the original restless project, we add a *calibrating project* having a single state, which we denote by $*$. The server must *engage* one of the two projects at each time, with the interpretation that the calibrating project is engaged when the original project is rested. As discussed in Section 4.3, we shall identify a project state e with a corresponding *job class* $e \in N^* = N \cup \{*\}$. We shall further associate with each policy $u \in \mathcal{U}$ performance measures $x_e(u)$, for $e \in N^*$, given by

$$x_e(u) = \begin{cases} x_e^0(u) & \text{if } e \in N; \\ \sum_{e' \in N} \theta_{e'} x_{e'}^1(u) & \text{if } e = *. \end{cases} \quad (41)$$

Note that performance measure $x_*(u)$ represents the total expected discounted number of successes, whereas $x_e(u)$ is the total expected discounted number of times the original project is rested in state e , for $e \in N$, under policy u . In this setting, the notion of service *priority* required by the PCL framework is the natural one for the project scheduling model just described. Note that the only relevant relative priorities are those between class $*$ and each class $e \in N$, which correspond to relative priorities between the two projects.

In this way, we can reformulate problem (32) (*normalized* so that $\mathbf{C}^1 = \mathbf{0}$) as

$$Z^{\text{OPT}}(\gamma; \mathbf{C}^0, \mathbf{0}) = \min_{u \in \mathcal{U}} \gamma x_*(u) + \sum_{e \in N} C_e^0 x_e(u). \quad (42)$$

Formulation (42) will allow us to apply later the index decomposition result in Theorem 8.

5.4 Decomposition laws

The main result in this section, Lemma 12 below, is the key to our formulation of PCL for system performance measures $x_e(u)$, for $e \in N^*$. It formulates a family of *decomposition laws*, i.e., a family of linear equations—having a certain special structure—relating active and passive performance measures. We use the term decomposition laws because such equations are analogous to the *work decomposition laws* satisfied by certain multiclass queueing systems (cf. Theorem 5 in [3]).

Lemma 12 (Decomposition laws) For any policy $u \in \mathcal{U}$ and state/class subset $S \in 2^N$,

$$x_*(u) + \sum_{e \in S} A_e^S x_e(u) = \sum_{e \in N} p_e V_e^S + \sum_{e \in S^c} A_e^S x_e^1(u); \quad (43)$$

in particular, when $S = N$,

$$x_*(u) + \sum_{e \in N} A_e^N x_e(u) = \sum_{e \in N} p_e V_e^N.$$

Proof

To simplify notation, we shall write $\mathbf{x}^a(u) = \mathbf{x}^a$, for $a \in \{0, 1\}$, and consider the \mathbf{x}^a 's to be row vectors. From the standard LP formulation given in (33), we consider the system of linear identities (34) satisfied by these performance measures, and rewrite it, for a given subset $S \in 2^N$, as

$$\begin{bmatrix} \mathbf{x}_S^0 & \mathbf{x}_{S^c}^0 \end{bmatrix} \begin{bmatrix} \mathbb{I}_S - \beta \mathbb{P}_{SS}^0 & -\beta \mathbb{P}_{SS^c}^0 \\ -\beta \mathbb{P}_{S^cS}^0 & \mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^0 \end{bmatrix} + \begin{bmatrix} \mathbf{x}_S^1 & \mathbf{x}_{S^c}^1 \end{bmatrix} \begin{bmatrix} \mathbb{I}_S - \beta \mathbb{P}_{SS}^1 & -\beta \mathbb{P}_{SS^c}^1 \\ -\beta \mathbb{P}_{S^cS}^1 & \mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_S & \mathbf{p}_{S^c} \end{bmatrix},$$

or, equivalently,

$$\begin{aligned} \mathbf{x}_S^0 (\mathbb{I}_S - \beta \mathbb{P}_{SS}^0) &= \mathbf{p}_S + \beta \mathbf{x}_{S^c}^0 \mathbb{P}_{S^cS}^0 + \beta \mathbf{x}_{S^c}^1 \mathbb{P}_{S^cS}^1 - \mathbf{x}_S^1 (\mathbb{I}_S - \beta \mathbb{P}_{SS}^1) \\ \mathbf{x}_{S^c}^1 (\mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^1) &= \mathbf{p}_{S^c} + \beta \mathbf{x}_S^0 \mathbb{P}_{SS^c}^0 + \beta \mathbf{x}_S^1 \mathbb{P}_{SS^c}^1 - \mathbf{x}_{S^c}^0 (\mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^0). \end{aligned}$$

Solving for \mathbf{x}_S^0 in the first of the last two equations, and substituting for it in the second, yields

$$\begin{aligned} \mathbf{x}_{S^c}^1 \mathbb{B} &= \mathbf{p}_{S^c} + \beta \mathbf{p}_S (\mathbb{I}_S - \beta \mathbb{P}_{SS}^0)^{-1} \mathbb{P}_{SS^c}^0 \\ &\quad + \beta \mathbf{x}_S^1 [\mathbb{P}_{SS^c}^1 - (\mathbb{I}_S - \beta \mathbb{P}_{SS}^1) (\mathbb{I} - \beta \mathbb{P}_{SS}^0)^{-1} \mathbb{P}_{SS^c}^0] \\ &\quad - \mathbf{x}_{S^c}^0 [\mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^0 - \beta^2 \mathbb{P}_{S^cS}^0 (\mathbb{I}_S - \beta \mathbb{P}_{SS}^0)^{-1} \mathbb{P}_{SS^c}^0], \end{aligned}$$

where

$$\mathbb{B} = [\mathbb{I}_{S^c} - \beta \mathbb{P}_{S^cS^c}^1 - \beta^2 \mathbb{P}_{S^cS}^1 (\mathbb{I}_S - \beta \mathbb{P}_{SS}^0)^{-1} \mathbb{P}_{SS^c}^0]$$

Now, postmultiplying both sides of the above equation by $\mathbf{V}_{S^c}^{S^c}$, and simplifying the resulting expression using (36)–(40), and the identity (which follows from the definition of $\mathbf{A}_S^{S^c}$)

$$\mathbf{A}_S^{S^c} = \boldsymbol{\theta}_S + \beta [\mathbb{P}_{SS^c}^1 - (\mathbb{I}_S - \beta \mathbb{P}_{SS}^1) (\mathbb{I}_S - \beta \mathbb{P}_{SS}^0)^{-1} \mathbb{P}_{SS^c}^0] \mathbf{V}_{S^c}^{S^c},$$

we obtain

$$\mathbf{x}_{S^c}^1 \boldsymbol{\theta}_{S^c} = \mathbf{p} \mathbf{V}^{S^c} + \mathbf{x}_S^1 [\mathbf{A}_S^{S^c} - \boldsymbol{\theta}_S] - \mathbf{x}_{S^c}^0 \mathbf{A}_{S^c}^{S^c}.$$

i.e.,

$$\mathbf{x}^1 \boldsymbol{\theta} + \mathbf{x}_{S^c}^0 \mathbf{A}_{S^c}^{S^c} = \mathbf{p} \mathbf{V}^{S^c} + \mathbf{x}_S^1 \mathbf{A}_S^{S^c},$$

which, in extended form, yields (43). \square

The next result follows immediately from Lemma 12.

Corollary 1 For any state subset $S \in 2^N$, the following identities hold:

(a) under any policy $u \in \mathcal{U}$ that takes the passive action over states in S^c , i.e., that gives priority (in the equivalent project scheduling model) to class $*$ over S^c -jobs,

$$x_*(u) + \sum_{e \in S} A_e^S x_e(u) = \sum_{e \in N} p_e V_e^S; \quad (44)$$

(b) under any policy $u \in \mathcal{U}$ that takes the active action over states in S , i.e., that gives priority to S -jobs over class $*$,

$$\sum_{e \in S} A_e^S x_e(u) = 0. \quad (45)$$

5.5 PCL for two-project scheduling reformulation

In this section we identify conditions on model parameters under which performance vector $\mathbf{x}(u) = (x_e(u))_{e \in N^*}$, for the two-project scheduling problem above, satisfies an appropriate set of PCL.

Given the greedoid (N, \mathcal{F}) and weight matrix \mathbb{A} discussed above, let us write

$$\mathbb{A}_{N\mathcal{F}} = (A_e^S)_{e \in N, S \in \mathcal{F}},$$

so that $(N, \mathcal{F}; \mathbb{A}_{N\mathcal{F}})$ is a weighted greedoid (see Definition 1). Recall that the two-project scheduling problem above can be seen as a scheduling problem over class set $N^* = \{*\} \cup N$. Let us further define $\mathcal{F}^* \subseteq 2^{N^*}$ by

$$\mathcal{F}^* = \{S^* = S_1 \cup S_2 : S_1 \in 2^{\{*\}}, S_2 \in \mathcal{F}\},$$

and let $A_e^{\{*\} \cup S}$ and A_*^S , for $e \in N^*$ and $S \in \mathcal{F}$, be given by

$$A_e^{\{*\} \cup S} = \begin{cases} A_e^S & \text{if } e \in N; \\ 1 & \text{if } e = *, \end{cases} \quad (46)$$

and $A_*^S = 1$. This defines weight matrices $\mathbb{A}_{N^*\mathcal{F}^*} = (A_e^S)_{e \in N^*, S \in \mathcal{F}^*}$ and $\mathbb{A}_{\{*\}2^{\{*\}}} = (A_*^S)_{S \in 2^{\{*\}}} = (1, 1)$. Note that, by construction, $(N^*, \mathcal{F}^*; \mathbb{A}_{N^*\mathcal{F}^*})$ is a weighted greedoid, which is *decomposed* (cf. Sections 3.3 and 4.3) into weighted greedoids $(N, \mathcal{F}; \mathbb{A}_{N\mathcal{F}})$ and $(\{*\}, 2^{N^*}; \mathbb{A}_{\{*\}2^{N^*}})$, so that Assumption 1 in Section 3.3 holds.

We further define $b(S^*)$, for $S^* \in \mathcal{F}^*$, by

$$b(S^*) = \begin{cases} \sum_{e \in N} p_e V_e^S & \text{if } S^* = \{*\} \cup S, \emptyset \neq S \in \mathcal{F} \\ 0 & \text{otherwise.} \end{cases}$$

Consider now the conditions on weights A_e^S given next.

Assumption 2 *Weight matrix $\mathbb{A}_{N\mathcal{F}} = (A_e^S)_{e \in N, S \in \mathcal{F}}$ satisfies*

$$\begin{aligned} A_e^S &> 0, & e \in S, S \in \mathcal{F} \\ A_e^S &\geq 0, & e \in S^c, S \in \mathcal{F}. \end{aligned}$$

Theorem 13 (PCL: discounted restless bandits) *Under Assumption 2, performance measures $x_*(u), x_e(u)$, $e \in N$, satisfy the PCL in Definition 7 with respect to weighted greedoid $(N^*, \mathcal{F}^*; \mathbb{A}_{N^*\mathcal{F}^*})$.*

Proof

The result follows immediately by combining the decomposition laws in Lemma 12 with the identities in Corollary 1 and the inequalities on the A_e^S in Assumption 2. \square

Suppose now that Assumption 2 holds. Let $(FEASIBLE, \gamma, \mathbf{e})$ be an output of adaptive-greedy algorithm $AG(\cdot | \mathbb{A}_{N\mathcal{F}}, N, \mathcal{F})$ corresponding to input $\mathbf{C}^0 = (C_e^0)_{e \in N}$. Assume that $FEASIBLE = TRUE$, so that passive cost vector \mathbf{C}^0 belongs in the corresponding *feasible cost region* \mathcal{C} (see Definition 4 in Section 3), and $\gamma = (\gamma_e)_{e \in N}$ is the vector of *allocation indices* in Definition 5. See Theorem 2. Note that it is immediate that the single allocation index for state $*$ corresponding to cost γ for weighted greedoid $(\{*\}, 2^{N^*}; \mathbb{A}_{\{*\}2^{N^*}})$ is precisely $\gamma_* = \gamma$.

Corollary 2 (θ -indexability) *If $\mathbf{C}^0 \in \mathcal{C}$ (i.e., $FEASIBLE = TRUE$), then problem (31) (normalized so that $\mathbf{C}^1 = \mathbf{0}$) is θ -indexable. Furthermore, the optimal indices are precisely the γ_e 's produced by adaptive-greedy algorithm $AG(\cdot | \mathbb{A}_{N\mathcal{F}}, N, \mathcal{F})$ on input \mathbf{C}^0 .*

Proof

The result follows by combining Theorem 13 with the decomposition result in Theorem 11. \square

We note that all of the above results, obtained for the discounted criterion, can be easily extended to the time-average criterion, by the same limiting argument (as $\beta \rightarrow 1$) developed in [13].

6 A simple restless bandit example

In this section we illustrate the approach developed in the paper through a simple single restless bandit model, motivated by a queueing input control problem (cf. [14]).

Let us consider a single restless bandit, as in Section 5, with state space $N = \{0, 1, 2\}$, active transition probability matrix given by

$$\mathbb{P}^1 = \begin{pmatrix} 1 - \lambda_0 & \lambda_0 & 0 \\ \mu_1 & 1 - \lambda_1 - \mu_1 & \lambda_1 \\ 0 & \mu_2 & 1 - \mu_2 \end{pmatrix},$$

and passive transition probability matrix

$$\mathbb{P}^0 = \begin{pmatrix} 1 & 0 & 0 \\ \mu_1 & 1 - \mu_1 & 0 \\ 0 & \mu_2 & 1 - \mu_2 \end{pmatrix}.$$

and initial state vector

$$\mathbf{p} = (p_0 \quad p_1 \quad p_2).$$

The immediate costs are of the form $C_i^0 = C_i^1 = C_i$.

We shall consider the following data:

$$C_0 = 0, C_1 = 2 - \mu_1, C_2 = 5 - \mu_2, \lambda_0 = 1/2, \lambda_1 = 1/3, \mu_1 = 1/2, \mu_2 = 2/3,$$

while the discount factor $0 < \beta < 1$ will remain general.

As for the policy family of interest, we shall consider it is characterized by the greedoid $\mathcal{F} = \{\emptyset, \{0\}, \{0, 1\}, \{0, 1, 2\}\}$, which defines a *nested* family of possible active sets.

The first step in the analysis involves normalizing the cost coefficients, as discussed in Section 5: an equivalent problem is obtained by letting all active costs to be zero, and redefining the passive cost vector as:

$$\widehat{\mathbf{C}}^0 = (\mathbb{I} - (\mathbb{I} - \beta \mathbb{P}^0) (\mathbb{I} - \beta \mathbb{P}^1)^{-1}) \mathbf{C} = \begin{pmatrix} \frac{1}{6} (27 + 17\beta) \frac{\beta}{\beta^2 - 6} \\ \frac{1}{6} (9\beta + 34) \frac{\beta}{\beta^2 - 6} \\ 0 \end{pmatrix}.$$

The next step involves computing the V_e^S 's and the A_e^S 's, with respect to give success probability parameters $\theta_0, \theta_1, \theta_2$. We thus obtain

$$\begin{pmatrix} V_0^{\{0\}} \\ V_1^{\{0\}} \\ V_2^{\{0\}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \frac{2-\beta}{1-\beta} \theta_0 \\ \frac{1}{2} \frac{\beta}{1-\beta} \theta_0 \\ \frac{\beta^2}{(1-\beta)(3-\beta)} \theta_0 \end{pmatrix},$$

$$\begin{pmatrix} A_0^{\{0\}} \\ A_1^{\{0\}} \\ A_2^{\{0\}} \end{pmatrix} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} + \beta \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 \\ 0 & -\lambda_1 & \lambda_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_0^{\{0\}} \\ V_1^{\{0\}} \\ V_2^{\{0\}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} (2 - \beta) \theta_0 \\ \frac{1}{2} \frac{2\theta_1(3-\beta) - \beta^2\theta_0}{3-\beta} \\ \theta_2 \end{pmatrix},$$

$$\begin{pmatrix} V_0^{\{0,1\}} \\ V_1^{\{0,1\}} \\ V_2^{\{0,1\}} \end{pmatrix} = \frac{1}{(1-\beta)(6-\beta^2)} \begin{pmatrix} (6-3\beta-\beta^2)\theta_0 + (3\beta-\beta^2)\theta_1 \\ (3-\beta)(\beta\theta_0 + (2-\beta)\theta_1) \\ 2\beta(\beta\theta_0 + (2-\beta)\theta_1) \end{pmatrix},$$

$$\begin{pmatrix} A_0^{\{0,1\}} \\ A_1^{\{0,1\}} \\ A_2^{\{0,1\}} \end{pmatrix} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} + \beta \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 \\ 0 & -\lambda_1 & \lambda_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_0^{\{0,1\}} \\ V_1^{\{0,1\}} \\ V_2^{\{0,1\}} \end{pmatrix} = \begin{pmatrix} \frac{(6-3\beta-\beta^2)\theta_0 + (3\beta-\beta^2)\theta_1}{6-\beta^2} \\ \frac{-\beta^2\theta_0 + (6-2\beta)\theta_1}{6-\beta^2} \\ \theta_2 \end{pmatrix},$$

$$\begin{pmatrix} V_0^{\{0,1,2\}} \\ V_1^{\{0,1,2\}} \\ V_2^{\{0,1,2\}} \end{pmatrix} = \frac{1}{(1-\beta)(6-\beta^2)} \begin{pmatrix} (6-3\beta-\beta^2)\theta_0 + (3\beta-\beta^2)\theta_1 + \beta^2\theta_2 \\ (3\beta-\beta^2)\theta_0 + (6-5\beta+\beta^2)\theta_1 + (2\beta-\beta^2)\theta_2 \\ 2\beta^2\theta_0 + (4\beta-2\beta^2)\theta_1 + (6-4\beta-\beta^2)\theta_2 \end{pmatrix},$$

and

$$\begin{pmatrix} A_0^{\{0,1,2\}} \\ A_1^{\{0,1,2\}} \\ A_2^{\{0,1,2\}} \end{pmatrix} = \begin{pmatrix} \frac{(6-3\beta-\beta^2)\theta_0 + (3\beta-\beta^2)\theta_1 + \beta^2\theta_2}{6-\beta^2} \\ \frac{-\theta_0\beta^2 + (6-2\beta)\theta_1 + 2\beta\theta_2}{6-\beta^2} \\ \theta_2 \end{pmatrix}.$$

Let us take $\theta_0 = \lambda_0$, $\theta_1 = \lambda_1$ and $\theta_2 = 0$. Then,

$$A_0^{\{0\}} = \frac{1}{2} - \frac{1}{4}\beta > 0, \quad A_1^{\{0\}} = \frac{1}{12} \frac{12 - 4\beta - 3\beta^2}{3 - \beta} > 0, \quad A_2^{\{0\}} = 0$$

$$A_0^{\{0,1\}} = \frac{1}{6} \frac{18 - 3\beta - 5\beta^2}{6 - \beta^2} > 0, \quad A_1^{\{0,1\}} = \frac{1}{6} \frac{12 - 4\beta - 3\beta^2}{6 - \beta^2} > 0, \quad A_2^{\{0,1\}} = 0$$

$$A_0^{\{0,1,2\}} = \frac{3 - \frac{1}{2}\beta - \frac{5}{6}\beta^2}{6 - \beta^2} > 0, \quad A_1^{\{0,1,2\}} = \frac{2 - \frac{2}{3}\beta - \frac{1}{2}\beta^2}{6 - \beta^2} > 0, \quad A_2^{\{0,1,2\}} = 0$$

Consider the performance measures:

$$\begin{aligned} x_*(u) &= \theta_0 x_0^1(u) + \theta_1 x_1^1(u) \\ x_0(u) &= x_0^0(u) \\ x_1(u) &= x_1^0(u). \end{aligned}$$

The parametric problem family of interest (cf. (32)) is

$$\min_{u \in \mathcal{U}} \gamma x_*(u) + \widehat{C}_0^0 x_0^0(u) + \widehat{C}_1^0 x_1^0(u) \quad (47)$$

Now, the above inequalities on the A_e^S ensure that Assumption 2 holds. Therefore, Theorem 13 guarantees that performance measures $x_*(u)$, $x_0^0(u)$, $x_1^0(u)$ satisfy PCL with respect to the weighted greedoid having ground set $N^* = \{*, 0, 1\}$, feasible set family

$$\mathcal{F}^* = \{S_1 \cup S_2 : S_1 \in 2^{\{*\}}, S_2 \in \mathcal{F}\},$$

and weights A_e^S as defined above, further extended as in (46). Therefore, Corollary 2 applies, establishing the θ -indexability of the problem, and further the optimality of \mathcal{F} -policies defined by the corresponding indices γ_i , under *feasible* cost vectors, as defined by the adaptive-greedy algorithm.

We next show that the given costs are feasible, and compute the corresponding allocation indices.

We have

$$y^{\{0,1\}} = \gamma_1 = \frac{\widehat{C}_1^0}{A_1^{\{0,1\}}} = \beta \frac{9\beta + 34}{-12 + 4\beta + 3\beta^2},$$

$$y^{\{0,1\}} - \frac{\widehat{C}_0^0}{A_0^{\{0,1\}}} = \frac{-6(\beta - 8)(\beta^2 - 6)\beta}{(-18 + 3\beta + 5\beta^2)(-12 + 4\beta + 3\beta^2)} < 0,$$

$$y^{\{0\}} = \frac{\widehat{C}_0^0 - A_0^{\{0,1\}}y^{\{0,1\}}}{A_0^{\{0\}}} = 4\beta \frac{8 - \beta}{(2 - \beta)(12 - 4\beta - 3\beta^2)} > 0$$

and

$$\gamma_0 = \gamma_1 + y^{\{0\}} = -3 \frac{\beta}{2 - \beta}.$$

Therefore, given an arbitrary $\gamma \in \mathbb{R}$, an optimal policy for problem (47) is given as follows: take the active action in state $i \leq 1$ having $\gamma_i > \gamma$, and the passive action otherwise. Note that the action taken in state 2 is irrelevant in this example.

References

- [1] Bertsimas, D. and Niño-Mora, J. (1996). Conservation laws, extended polymatroids and multiarmed bandit problems; a unified approach to indexable systems. *Math. Oper. Res.* **21** 257–306.
- [2] Bhattacharya, P. P., Georgiadis, L. and Tsoucas, P. (1991). Extended polymatroids: properties and optimization. In *Proceedings of First International Conference on Integer Programming and Combinatorial Optimization, Carnegie Mellon University*, pp. 298–315.
- [3] Bertsimas, D. and Niño-Mora, J. (1999). Optimization of multiclass queueing networks with changeover times via the achievable region method: Part I, the single-station case. *Math. Oper. Res.* **24** 306–330.
- [4] Bertsimas, D. and Niño-Mora, J. (2000). Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Oper. Res.* vol. **48** no. 1 (in press).
- [5] Coffman, E. G., Jr. and Mitran, I. (1980). A characterization of waiting time performance realizable by single-server queues. *Oper. Res.* **28**, part 2, 810–821.

- [6] Dacre, M., Glazebrook, K.D. and Niño-Mora, J. (1999). The achievable region approach to the optimal control of stochastic systems (with discussion and a reply by the authors). *J. Roy. Statist. Soc. Ser. B* **61** 747–791.
- [7] Edmonds, J. (1970). Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, 69–87, Gordon and Breach, New York.
- [8] Federgruen, A. and Groenevelt, H. (1988). Characterization and optimization of achievable performance in general queueing systems. *Oper. Res.* **36** 733–741.
- [9] Garbe, R. and Glazebrook, K. D. (1998). Stochastic scheduling with priority classes. *Math. Oper. Res.* **23** 119–144.
- [10] Gittins, J. C. (1979). Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Statist. Soc. Ser. B* **41** 148–177.
- [11] Klimov, G.P. (1974). Time sharing service systems I. *Theory Probab. Appl.* **19** 532–551.
- [12] Korte, B. and Lovász, L. (1981). Mathematical structures underlying greedy algorithms. In *Fundamentals of Computation Theory, Lecture Notes in Computer Science*, vol. 117, F. Grecseg (Ed.), Springer, Berlin-New York, pp. 205–209.
- [13] Niño-Mora, J. (1999). Restless bandits, partial conservation laws and indexability. Working Paper no. 435, Dept. of Economics and Business, Universitat Pompeu Fabra. Available at <http://www.econ.upf.es/cgi-bin/onepaper?435>. Submitted to *Advances in Applied Probability*.
- [14] Niño-Mora, J. (2000). Optimal arrival control to a queueing system, partial conservation laws, and restless bandit allocation indices. Working paper, Dept. of Economics and Business, Universitat Pompeu Fabra.
- [15] Niño-Mora, J. (2000). Design and analysis of structured index policies in stochastic scheduling: a primal-dual approach via greedoid polyhedra. Working paper, Dept. of Economics and Business, Universitat Pompeu Fabra. Short version submitted to AP-PROX 2000.
- [16] Papadimitriou, C. H. and Tsitsiklis, J. N. (1999). The complexity of optimal queueing network control. *Math. Oper. Res.* **24** 293–305.
- [17] Puterman, M.L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York.
- [18] Shanthikumar, J. G. and Yao, D. D. (1992). Multiclass queueing systems: polymatroidal structure and optimal scheduling control. *Oper. Res.* **40** suppl. 2, S293–S299.
- [19] Tsoucas, P. (1991). The region of achievable performance in a model of Klimov. Research Report RC16543, IBM T. J. Watson Research Center, Yorktown Heights, NY.
- [20] Weber, R. R. and Weiss, G. (1990). On an index policy for restless bandits. *J. Appl. Prob.* **27** 637–648.
- [21] Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. In *A Celebration of Applied Probability*, J. Gani (Ed.), *J. Appl. Prob.* **25A** 287–298.