Conservation Laws, Extended Polymatroids and Multi-armed Bandit Problems; a Unified Approach to Indexable Systems

Dimitris Bertsimas, Jose Niño-Mora

OR 277-93 March 1993

Conservation laws, extended polymatroids and multi-armed bandit problems; a unified approach to indexable systems

Dimitris Bertsimas [1] Jose Niño-Mora [2]

February 1993

[1] Dimitris Bertsimas, Sloan School of Management and Operations Research Center, MIT, Cambridge, MA 02139. The research of the author was partially supported by a Presidential Young Investigator Award DDM-9158118 with matching funds from Draper Laboratory

[2] Jose Niño-Mora, Operations Research Center, MIT, Cambridge, MA 02139. The research of the author was partially supported by a MEC/Fulbright Fellowship.

Abstract

We show that if performance measures in stochastic and dynamic scheduling problems satisfy generalized conservation laws, then the feasible space of achievable performance is a polyhedron called an extended polymatroid that generalizes the usual polymatroids introduced by Edmonds. Optimization of a linear objective over an extended polymatroid is solved by an adaptive greedy algorithm, which leads to an optimal solution having an indexability property (*indexable systems*). Under a certain condition, then the indices have a stronger decomposition property (*decomposable systems*). The following classical problems can be analyzed using our theory: multi-armed bandit problems, branching bandits. multiclass queues, multiclass queues with feedback, deterministic scheduling problems. Interesting consequences of our results include: (1) a characterization of indexable systems as systems that satisfy generalized conservation laws, (2) a sufficient condition for indexable systems to be decomposable, (3) a new linear programming proof of the decomposability property of Gittins indices in multi-armed bandit problems, (4) a unified and practical approach to sensitivity analysis of indexable systems, (5) a new characterization of the indices of indexable systems as sums of dual variables and a new interpretation of the indices in terms of retirement options in the context of branching bandits, (6) the first rigorous analysis of the indexability of undiscounted branching bandits, (7) a new algorithm to compute the indices of indexable systems (in particular Gittins indices), which is as fast as the fastest known algorithm, (8) a unification of the algorithm of Klimov for multiclass queues and the algorithm of Gittins for multi-armed bandits as special cases of the same algorithm. (9) closed form formulae for the performance of the optimal policy, and (10) an understanding of the nondependence of the indices on some of the parameters of the stochastic scheduling problem. Most importantly, our approach provides a unified treatment of several classical problems in stochastic and dynamic scheduling and is able to address in a unified way their variations such as: discounted versus undiscounted cost criterion, rewards versus taxes. preemption versus nonpreemption, discrete versus continuous time, work conserving versus idling policies, linear versus nonlinear objective functions.

1 Introduction

In the mathematical programming tradition researchers and practitioners solve optimization problems by defining decision variables and formulating constraints, thus describing the feasible space of decisions, and applying algorithms for the solution of the underlying optimization problem. For the most part, the tradition for stochastic and dynamic scheduling problems has been, however, quite different, as it relies primarily on dynamic programming formulations. Using ingenious but often ad hoc methods, which exploit the structure of the particular problem, researchers and practitioners can sometimes derive insightful structural results that lead to efficient algorithms. In their comprehensive survey of deterministic scheduling problems Lawler et. al. [23] end their paper with the following remarks: "The results in stochastic scheduling are scattered and they have been obtained through a considerable and sometimes dishearting effort. In the words of Coffman, Hofri and Weiss [8], there is great need for new mathematical techniques useful for simplifying the derivation of the results".

Perhaps one of the most important successes in the area of stochastic scheduling in the last twenty years is the solution of the celebrated *multi-armed bandit problem*, a generic version of which in discrete time can be described as follows:

The multi-armed bandit problem: There are K parallel projects, indexed k = 1, ..., K. Project k can be in one of a finite number of states i_k. At each instant of discrete time $t = 0, 1, \dots$ one can work on only a single project. If one works on project k in state $i_k(t)$ at time t, then one receives an immediate expected reward of $R_{i_k(t)}$. Rewards are additive and discounted in time by a factor $0 < \beta < 1$. The state $i_k(t)$ changes to $i_k(t+1)$ by a Markov transition rule (which may depend on k, but not on t), while the states of the projects one has not engaged remain unchanged, i.e., $i_l(t+1) = i_l(t)$ for $l \neq k$. The problem is how to allocate one's resources sequentially in time in order to maximize expected total discounted reward over an infinite horizon.

The problem has numerous applications and a rather vast literature (see Gittins [16] and the references therein). It was originally solved by Gittins and Jones [14], who proved that to each project k one could attach an *index* $\gamma^k(i_k(t))$, which is a function of the project k and the current state $i_k(t)$ alone, such that the optimal action at time t is to engage the project of largest current index. They also proved the important result that these index functions satisfy a stronger *index decomposition* property: the function $\gamma^k(\cdot)$ only depends on characteristics of project k (states, rewards and transition probabilities), and not on any other project. These indices are now known as Gittins indices, in recognition of Gittins contribution. Since the original solution, which relied on an interchange argument, other proofs were proposed: Whittle [36] provided a proof based on dynamic programming, subsequently simplified by Tsitsiklis [30]. Varaiya, Walrand and Buyukkoc [33] and Weiss [35] provided different proofs based on interchange arguments. Weber [34] outlined an intuitive proof. More recently, Tsitsiklis [31] has provided a proof based on a simple inductive argument.

The multi-armed bandit problem is a special case of a dynamic and stochastic *job scheduling system S*. In this context, there is a set E of job types and we are interested in optimizing a function of a performance measure (rewards or taxes) under a class of *admissible scheduling policies.*

Definition 1 (Indexable Systems) We say that a dynamic and stochastic *job scheduling system S* is *indexable* if the following policy is optimal: To each job type i we attach an index, γ_i. At each decision epoch select a job with the largest index.

In general the indices γ_i could depend on the entire set E of job types. Consider a partition of the set E to subsets E_k, $k = 1, \ldots K$, which contain collections of job types and can be interpreted as projects consisting of several job types. In certain situations, the index of job type $i \in E_k$ depends only on the characteristics of the job types in E_k and not on the entire set E of job types. Such a property is particularly useful computationally since it enables the system to be decomposed to smaller parts and the computation of the indices can be done independently. As we have seen the multi-armed bandit problem has this decomposition property, which motivates the following definition:

Definition 2 (Decomposable Systems) An indexable system is called decomposable if for all job types $i \in E_k$, the index γ_i of job type i depends only on the characteristics of the set of job types E_k.

In addition to the multi-armed bandit problem, a variety of dynamic and stochastic scheduling problems has been solved in the last decades by indexing rules:

1. Extensions of the usual multi-armed bandit problem such as arm-acquiring bandits (Whittle [37], [38]) and more generally branching bandits (Weiss [35]), that include several important problems as special cases.

2. The multiclass queueing scheduling problem with Bernoulli feedback (Klimov [22], Tcha and Pliska [29]).

3. The multiclass queueing scheduling problem without feedback (Cox and Smith [9], Harrison [19], Kleinrock [21], Gelenbe and Mitrani [13], Shantikumar and Yao [26]).

4. Deterministic scheduling problems (Smith [27]).

An interesting distinction, which is not emphasized in the literature, is that examples (1) and (2) above are *indexable systems*, but they are not in general *decomposable systems*. Example (3), however, has a more refined structure. It is indexable, but not decomposable, under discounting, while it is decomposable under the average cost criterion (the $c\mu$ rule). As already observed, the multi-armed bandit problem is an example of a decomposable system, while example (4) above is also decomposable.

Faced with these results, one asks what is the underlying *deep reason* that these non-trivial problems have very efficient solutions both theoretically as well as practically. In particular, what is the class of stochastic and dynamic scheduling problems that are indexable? Under what conditions, indexable systems are decomposable? But most importantly is there a unified way to address stochastic and dynamic scheduling problems that will lead to a deeper understanding of their strong structural properties? This is the set of questions that motivates this work.

In the last decade the following approach has been proposed to address special cases of these questions. In broad terms, researchers try to describe the feasible space of a stochastic and dynamic scheduling problem as a polyhedron. Then, the stochastic and dynamic scheduling problem is translated to an optimization problem over the corresponding polyhedron, which can then be attacked by traditional mathematical programming methods. Coffman and Mitrani [7] and Gelenbe and Mitrani [13] first showed using conservation laws that the performance space of a multiclass queue under the average cost criterion can be described as a polyhedron. Federgruen and Groenevelt [11], [12] advanced the theory further by observing that in certain special cases of multiclass queues, the polyhedron has a very special structure (it is a polymatroid) that gives rise to very simple optimal policies (the $c\mu$ rule). Shantikumar and Yao [26] generalized the theory further by observing that if a system satisfies strong conservation laws, then the underlying performance space is necessarily a *polymatroid*. They also proved that, when the cost is linear on the performance, the optimal policy is a *fixed priority rule* (also called *head of the line* priority rule; see Cobham [6], and Cox and Smith [9]). Their results partially extend to some rather restricted queueing networks, in which they assume that all the different classes of customers have the same routing probabilities, and the same service requirements at each station of the network (see also [25]). Tsoucas ([32]) derived the region of achievable performance in the problem of scheduling a multiclass nonpreemptive M/G/1 queue with Bernoulli feedback, introduced by Klimov ([22]). Finally, Bertsimas *et al.* [2] generalize the ideas of conservation laws to general multiclass queueing networks using potential function ideas. They find linear and nonlinear inequalities that the feasible region satisfies. Optimization over this set of constraints gives bounds on achievable performance.

Our goal in this paper is to propose *a unified theory* of conservation laws and to establish that the very strong structural properties in the optimization of a class of stochastic and dynamic systems that include the multi-armed bandit problem and its extensions follow from the corresponding strong structural properties of the underlying polyhedra that characterize the regions of achievable performance.

By generalizing the work of Shantikumar and Yao [26] we show that if performance measures in stochastic and dynamic scheduling problems satisfy *generalized conservation laws*, then the feasible space of achievable performance is a polyhedron called an *extended polymatroid* (see Bhattacharya *et al.* [4]). Optimization of a linear objective over an extended polymatroid is solved by an adaptive greedy algorithm, which leads to an optimal solution having an indexability property. Special cases of our theory include all the problems we have mentioned, i.e., multi-armed bandit problems, discounted and undiscounted branching bandits, multiclass queues, multiclass queues with feedback and deterministic scheduling problems. Interesting consequences of our results include:

1. A characterization of indexable systems as systems that satisfy generalized conservation laws.

2. Sufficient conditions for indexable systems to be decomposable.

3. A genuinely new, algebraic proof (based on the strong duality theory of linear programming as opposed to dynamic programming formulations) of the decomposability property of Gittins indices in multi-armed bandit problems.

4. A unified and practical approach to sensitivity analysis of indexable systems, based on the well understood sensitivity analysis of linear programming.

5. A new characterization of the indices of indexable systems as sums of dual variables corresponding to the extended polymatroid that characterizes the feasible space.

6. A new interpretation of indices in the context of branching bandits as retirement options, thus generalizing the interpretation of Whittle [36] and Weber [34] for the indices of the classical multi-armed bandit problem.

7. The first complete and rigorous analysis of the indexability of undiscounted branching bandits.

8. A new algorithm to compute the indices of indexable systems (in particular Gittins indices), which is as fast as the fastest known algorithm (Varaiya, Walrand and Buyukkoc [33]).

9. The realization that the algorithm of Klimov for multiclass queues and the algorithm of Gittins for multi-armed bandits are examples of the same algorithm.

10. Closed form formulae for the performance of the optimal policy. This also leads to an understanding of the nondependence of the indices on some of the parameters of the stochastic scheduling problem.

Most importantly, our approach provides a unified treatment of several classical problems in stochastic and dynamic scheduling and is able to address in a unified way their variations such as: discounted versus undiscounted cost criterion, rewards versus taxes, preemption versus nonpreemption, discrete versus continuous time, work conserving versus idling policies, linear versus nonlinear objective functions.

The paper is structured as follows: In Section 2 we define the notion of generalized conservation laws and show that if a performance vector of a stochastic and dynamic scheduling problem satisfies generalized conservation laws, then the feasible space of this performance vector is an extended polymatroid. Using the duality theory of linear programming we show that linear optimization problems over extended polymatroids can be solved by an adaptive greedy algorithm. Most importantly, we show that this optimization problem has an indexability property. In this way, we give a characterization of indexable systems as systems that satisfy generalized conservation laws. We also find a sufficient condition for an indexable system to be decomposable and prove a powerful result on sensitivity analysis. In Section 3 we study a natural generalization of the classical multi-armed bandit problem: the branching bandit problem. We propose two different performance measures and prove that they satisfy generalized conservation laws, and thus from the results of the previous section their feasible space is an extended polymatroid. We then consider different cost and reward structures on branching bandits, corresponding to the discounted and undiscounted case, and some transform results. Section 4 contains applications of the previous sections to various classical problems: multi-armed bandits, multi-class queueing scheduling problems with or without feedback and deterministic scheduling problems. The final section contains some thoughts on the field of optimization of stochastic systems.

2 Extended Polymatroids and Generalized Conservation Laws

2.1 Extended Polymatroids

Tsoucas [32] characterized the performance space of Klimov's problem (see Klimov [22]) as a polyhedron with a special structure, not previously identified in the literature. Bhattacharya *et al.* [4] called this polyhedron an *extended polymatroid* and proved some interesting properties of it. Extended polymatroids are a central structure for the results we present in this paper.

Let us first establish the notation we will use. Let $E = \{1, \ldots, n\}$ be a finite set. Let x denote a real n-vector, with components x_i, for $i \in E$. For $S \subseteq N$, let $S^c = E \setminus S$. and let $|S|$ denote the cardinality of S. Let 2^E denote the class of all subsets of E. Let $b: 2^E - \Re_+$ be a set function, that satisfies $b(\emptyset) = 0$. Let $A = (A_i^S)_{i \in E,\ S \subseteq E}$ be a matrix that satisfies

$$A_i^S > 0, \quad \text{for} \quad i \in S \qquad \text{and} \qquad A_i^S = 0. \quad \text{for} \quad i \in S^c, \qquad \text{for all } S \subseteq E. \qquad (1)$$

Let $\pi = (\pi_1, \ldots, \pi_n)$ be a permutation of E. For clarity of presentation, it is convenient to introduce the following additional notation. For an n-vector $x = (x_1, \ldots, x_n)^T$ let $x_\pi = (x_{\pi_1}, \ldots, x_{\pi_n})^T$. Let us write

$$b_\pi = (b(\{\pi_1\}), b(\{\pi_1, \pi_2\}), \ldots, b(\{\pi_1, \ldots, \pi_n\}))^T.$$

Let A_π denote the following lower triangular submatrix of A:

$$A_\pi = \begin{pmatrix} A_{\pi_1}^{\{\pi_1\}} & 0 & \dots & 0 \\ A_{\pi_1}^{\{\pi_1,\pi_2\}} & A_{\pi_2}^{\{\pi_1,\pi_2\}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{\pi_1}^{\{\pi_1,\dots,\pi_n\}} & A_{\pi_2}^{\{\pi_1,\dots,\pi_n\}} & \dots & A_{\pi_n}^{\{\pi_1,\dots,\pi_n\}} \end{pmatrix}.$$

Let $v(\pi)$ be the unique solution of the linear system

$$\sum_{i=1}^{j} A_{\pi_i}^{\{\pi_1,\dots,\pi_j\}} x_{\pi_i} = b(\{\pi_1,\dots,\pi_j\}), \quad j = 1,\dots,n \tag{2}$$

or, in matrix notation:

$$A_\pi x_\pi = b_\pi. \tag{3}$$

Let us define the polyhedron

$$\mathcal{P}(A,b) = \{\, x \in \Re^n : \sum_{i\in S} A_i^S x_i \geq b(S), \quad \text{for } S \subseteq E \,\} \tag{4}$$

and the polytope

$$\mathcal{B}(A,b) = \{\, x \in \Re^n : \sum_{i\in S} A_i^S x_i \geq b(S), \quad \text{for } S \subset E \quad \text{and} \quad \sum_{i\in E} A_i^E x_i = b(E)\}. \tag{5}$$

Note that if $x \in \mathcal{P}(A,b)$, then it follows that $x \geq 0$ componentwise. The following definition is due to Bhattacharya *et al.* [4].

Definition 3 (Extended Polymatroid) We say that the polyhedron $\mathcal{P}(A,b)$ is an ***extended polymatroid*** with base set E, if for every permutation π of E, $v(\pi) \in \mathcal{P}(A,b)$. In this case we say that the polytope $\mathcal{B}(A,b)$ is the *base* of the extended polymatroid $\mathcal{P}(A,b)$.

2.2 Optimization over Extended Polymatroids

Extended polymatroids are polyhedra defined by an exponential number of inequalities. Yet, Tsoucas [32] and Bhattacharya *et al.* [4] presented a polynomial algorithm, based on Klimov's algorithm (see Klimov [22]) for solving a linear programming problem over an extended polymatroid. In this subsection we provide a new duality proof that this algorithm solves the problem optimally. We then show that we can associate with this linear program certain *indices*, related to the dual program, in such a way that the problem has an *indexability* property. Under certain conditions, we prove that a stronger *index*

decomposition property holds. We also present an optimality condition specially suited for performing sensitivity analysis.

In what follows we assume that $\mathcal{P}(A, b)$ is an extended polymatroid. Let $R \in \Re^n$ be a row vector. Let us consider the following linear programming problem:

$$(P) \qquad \max\{ \sum_{i \in E} R_i x_i : x \in \mathcal{B}(A, b) \}. \tag{6}$$

Note that since $\mathcal{B}(A, b)$ is a polytope, this linear program has a finite optimal solution. Therefore we may consider its dual, and this will have the same optimum value. We shall have a dual variable y^S for every $S \subseteq E$. The dual problem is:

$$(D) \qquad \min\{ \sum_{S \subseteq E} b(S) y^S : \sum_{S \ni i} A_i^S y^S = R_i, \quad \text{for } i \in E, \quad \text{and} \quad y^S \leq 0, \quad \text{for } S \subset E \}. \tag{7}$$

In order to solve (P), Bhattacharya *et al.* [4] presented the following *adaptive greedy algorithm*, based on Klimov's algorithm [22]:

Algorithm $\mathcal{A}_1$

Input: (R, A).

Output: $(\pi, \overline{y}, \nu, \mathcal{S})$, where $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation of E, $\overline{y} = (\overline{y}^S)_{S \subseteq E}$, $\nu = (\nu_1, \ldots, \nu_n)$, and $\mathcal{S} = \{S_1, \ldots, S_n\}$, with $S_k = \{\pi_1, \ldots, \pi_k\}$, for $k \in E$.

Step 0. Set $S_n = E$. Set $\nu_n = \max\{ \frac{R_i}{A_i^E} : i \in E \}$;
pick $\pi_n \in \operatorname{argmax}\{ \frac{R_i}{A_i^E} : i \in E \}$.

Step k. For $k = 1, \ldots, n-1$:
Set $S_{n-k} = S_{n-k+1} \setminus \{\pi_{n-k+1}\}$; set $\nu_{n-k} = \max\{ \frac{R_i - \sum_{j=0}^{k-1} A_i^{S_{n-j}} \nu_{n-j}}{A_i^{S_{n-k}}} : i \in S_{n-k} \}$;
pick $\pi_{n-k} \in \operatorname{argmax}\{ \frac{R_i - \sum_{j=0}^{k-1} A_i^{S_{n-j}} \nu_{n-j}}{A_i^{S_{n-k}}} : i \in S_{n-k} \}$.

Step n. For $S \subseteq E$ set

$$\overline{y}^S = \begin{cases} \nu_j, & \text{if } S = S_j \text{ for some } j \in E; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that the complexity of $\mathcal{A}_1$, given (R, A), is $O(n^3)$. Note that, for certain reward vectors, ties may occur in algorithm $\mathcal{A}_1$. In the presence of ties, the permutation π generated depends clearly on the choice of tie-breaking rules. However, we will show that vectors ν and $\overline{y}$ are uniquely determined by $\mathcal{A}_1$. In order to prove this point, whose importance will be clear later, and to understand better $\mathcal{A}_1$, let us introduce the following related algorithm:

Algorithm $\mathcal{A}_2$

Input: (R, A).

Output: $(r, \overline{\overline{y}}, \mathcal{H}, \mathcal{J})$, where $1 \leq r \leq n$ is an integer, $\overline{\overline{y}} = (\overline{\overline{y}}^S)_{S \subseteq E}$, $\mathcal{H} = \{H_1, \ldots, H_r\}$ is a partition of E, and $J_k = \cup_{l=k}^{r} H_l$, for $l = 1, \ldots, r$.

```
Step 1. Set $k := 1$; set $J_1 = E$;
    set $\theta_1 = \max\{ \frac{R_i}{A_i^E} : i \in E \}$   and   $H_1 = \text{argmax}\{ \frac{R_i}{A_i^E} : i \in E \}$.

Step 2. While $J_k \neq H_k$ do:
    begin
    Set $k := k + 1$; set $J_k = J_{k-1} \setminus H_{k-1}$;
    set $\theta_k = \max\{ \frac{R_i - \sum_{l=1}^{k-1} A_i^{J_l} \theta_l}{A_i^{J_k}} : i \in J_k \}$   and   $H_k = \text{argmax}\{ \frac{R_i - \sum_{l=1}^{k-1} A_i^{J_l} \theta_l}{A_i^{J_k}} : i \in J_k \}$.
    end {while}

Step 3. Set $r = k$;
    for $S \subseteq E$ set
        $\overline{\overline{y}}^S = \begin{cases} \theta_k, & \text{if } S = J_k \text{ for some } k = 1, \ldots, r; \\ 0, & \text{otherwise.} \end{cases}$
```

In what follows let $(\pi, \overline{y}, \nu, \mathcal{S})$ be an output of $\mathcal{A}_1$ and let $(r, \overline{\overline{y}}, \mathcal{H}, \mathcal{J})$ be the output of $\mathcal{A}_2$. Note that the output of algorithm $\mathcal{A}_2$ is uniquely determined by its input.

The idea that algorithm $\mathcal{A}_2$ is just an unambiguous version of $\mathcal{A}_1$ is formalized in the following result:

Proposition 1 *The following relations hold between the outputs of algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$:*

(a) *for $l = 1, \ldots, n$*

$$\nu_l = \begin{cases} \theta_k, & \text{if } l = |J_k| \text{ for some } k = 1, \ldots, r; \\ 0, & \text{otherwise;} \end{cases} \tag{8}$$

(b) $\overline{y} = \overline{\overline{y}}$;

(c) π *satisfies*

$$J_k = \{\pi_1, \ldots, \pi_{|J_k|}\}, \quad k = 1, \ldots, r, \tag{9}$$

$$H_k = \{\pi_{|J_k|-|H_k|+1}, \ldots, \pi_{|J_k|}\}, \quad k = 1, \ldots, r. \tag{10}$$

Outline of the proof

Parts (a) and (c) follow by induction arguments. Part (b) follows by (a) and the definitions of $\overline{y}$ and $\overline{\overline{y}}$. □

Remark: Proposition 1 shows that $\overline{y}$ and ν are uniquely determined (and thus invariant under different tie-breaking rules) by algorithm $\mathcal{A}_1$. It also reveals in (c) the structure of the permutations π that can be generated by $\mathcal{A}_1$.

Tsoucas [32] and Bhattacharya *et al.* [4] proved from first principles that algorithm $\mathcal{A}_1$ solves linear program (P) optimally. Next we provide a new proof, using linear programming duality theory.

Proposition 2 *Let vector* $\overline{y}$ *and permutation* π *be generated by algorithm* $\mathcal{A}_1$. *Then* $x(\pi)$ *and* $\overline{y}$ *are an optimal primal-dual pair for the linear programs* (P) *and* (D).

Proof

We first show that $\overline{y}$ is dual feasible. By definition of ν_n in $\mathcal{A}_1$, it follows that

$$R_i - A_i^{S_n} \nu_n \leq 0, \quad i \in S_n$$

and since $S_{n-1} \subset S_n$ it follows that $\nu_{n-1} \leq 0$.

Similarly, for $k = 1, \ldots, n-2$, by definition of ν_{n-k} it follows that

$$R_i - \sum_{j=0}^{k} A_i^{S_{n-j}} \nu_{n-j} \leq 0, \quad i \in S_{n-k},$$

and since $S_{n-k-1} \subset S_{n-k}$, it follows that $\nu_{n-k-1} \leq 0$. Hence $\nu_j \leq 0$, for $j = 1, \ldots, n-1$, and by definition of $\overline{y}$, we have $\overline{y}^S \leq 0$, for $S \subset E$.

Moreover, for $k = 0, 1, \ldots, n-1$ we have, by construction,

$$\sum_{S \ni \pi_{n-k}} A_{\pi_{n-k}}^S \overline{y}^S = \sum_{j=0}^{k} A_{\pi_{n-k}}^{S_{n-j}} \nu_{n-j} = R_{\pi_{n-k}}.$$

Hence $\overline{y}$ is dual feasible.

Let $\overline{x} = v(\pi)$. Since $\mathcal{P}(A,b)$ is an extended polymatroid, $\overline{x}$ is primal feasible. Let us show that $\overline{x}$ and $\hat{y}$ satisfy complementary slackness. Assume $\overline{y}^S \neq 0$. Then, by construction it must be $S = S_k = \{\pi_1, \dots, \pi_k\}$, for some k. And since $\overline{x}$ satisfies (3), it follows that

$$\sum_{i \in S} A_i^S \overline{x}_i = \sum_{j=1}^{k} A_{\pi_j}^{\{\pi_1,\dots,\pi_k\}} \overline{x}_{\pi_j} = b(S).$$

Hence, by strong duality $v(\pi)$ and $\overline{y}$ are an optimal primal-dual pair, and this completes the proof. □

Remark: Edmonds [10] introduced a special class of polyhedra called *polymatroids*, and proved the classical result that the ***greedy*** algorithm solves the linear optimization problem over a polyhedron for every linear objective function if and only if the polyhedron is a polymatroid. Now, in the case that $A_i^S = 1$, for $i \in S$, and $S \subseteq E$, it is easy to see that $\mathcal{A}_1$ is the greedy algorithm that sorts the R_i's in nonincreasing order. By Edmond's result and Proposition 2 it follows that in this case $\mathcal{B}(A,b)$ is a polymatroid. Therefore, extended polymatroids are the natural generalizations of polymatroids, and algorithm $\mathcal{A}_1$ is the natural extension of the greedy algorithm.

The fact that $v(\pi)$ and $\overline{y}$ are optimal solutions has some important consequences. It is well known that every extreme point of a polyhedron is the unique maximizer of some linear objective function. Therefore, the $v(\pi)$'s are the only extreme points of $\mathcal{B}(A,b)$. Hence it follows:

Theorem 1 (Characterization of Extreme Points) *The set of extreme points of $\mathcal{B}(A,b)$ is*

$$\{\, v(\pi) : \pi \textit{ is a permutation of } E \,\}.$$

The optimality of the adaptive greedy algorithm $\mathcal{A}_1$ leads naturally to the definition of certain indices, which for historical reasons, that will be clear later, we call generalized Gittins indices.

Definition 4 (Generalized Gittins Indices) Let $\overline{y}$ be the optimal dual solution generated by algorithm $\mathcal{A}_1$. Let

$$\gamma_i = \sum_{S:\ E \supseteq S \ni i} \overline{y}^S, \quad i \in E. \tag{11}$$

We say that $\gamma_1, \dots, \gamma_n$ are the *generalized Gittins indices* of linear program (P).

Remark: Notice that by Proposition 1(a) and the definition of $\overline{y}$, it follows that if permutation π is an output of algorithm $\mathcal{A}_1$ then the generalized Gittins indices can be computed as follows:

$$\begin{aligned} \gamma_{\pi_i} &= \nu_n + \cdots + \nu_i && (12) \\ &= \overline{y}^{\{\pi_1,\ldots,\pi_n\}} + \cdots + \overline{y}^{\{\pi_1,\ldots,\pi_i\}}, \quad i \in E. && (13) \end{aligned}$$

Let $\Re_- = \{x \in \Re : x \leq 0\}$. Let $\gamma_1, \ldots, \gamma_n$ be the generalized Gittins indices of (P). Let π be a permutation of E. Let T be the following $n \times n$ lower triangular matrix:

$$T = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix}.$$

In the next proposition and the next theorem we reveal the equivalence between some optimality conditions for linear program (P).

Proposition 3 *The following statements are equivalent:*

(a) π *satisfies (9) and (10);*

(b) π *is an output of algorithm* $\mathcal{A}_1$*;*

(c) $R_\pi A_\pi^{-1} \in \Re_-^{n-1} \times \Re$, *and then the generalized Gittins indices are given by* $\gamma_\pi = R_\pi A_\pi^{-1} T$*;*

(d) $\gamma_{\pi_1} \leq \gamma_{\pi_2} \leq \cdots \leq \gamma_{\pi_n}$.

Outline of the proof

(a) $\Rightarrow$ (b): Proved in Proposition 1(a).

(b) $\Rightarrow$ (c): It is clear, by construction in $\mathcal{A}_1$, that

$$\nu = R_\pi A_\pi^{-1}. \qquad (14)$$

Now, in the proof of Proposition 2 we showed that $\nu \in \Re_-^{n-1} \times \Re$. Moreover, by (12) we get

$$\gamma_\pi = \nu T,$$

and by (14) it follows that

$$\gamma_\pi = R_\pi A_\pi^{-1} T.$$

(c) ⇒ (d): By (c) we have

$$\begin{pmatrix} \gamma_{\pi_1} - \gamma_{\pi_2} \\ \vdots \\ \gamma_{\pi_{n-1}} - \gamma_{\pi_n} \\ \gamma_{\pi_n} \end{pmatrix} = \gamma_\pi T^{-1} = R_\pi A_\pi^{-1} \in \Re_-^{n-1} \times \Re,$$

whence the result follows.

(d) ⇒ (a): By construction of $\overline{\overline{y}}$ in algorithm $\mathcal{A}_2$, the fact that $\overline{y} = \overline{\overline{y}}$ and the definition of the generalized Gittins indices, it follows that

$$\gamma_i = \theta_1 + \cdots + \theta_k, \qquad \text{for } i \in H_k, \quad \text{and } k = 1, \ldots, r. \tag{15}$$

Also, it is easy to see that $\theta_j < 0$, for $j \geq 2$. These two facts clearly imply that π must satisfy (10), and hence (9), which completes the proof of the proposition. □

Combining the result that algorithm $\mathcal{A}_1$ solves linear program (P) optimally with the equivalent conditions in Proposition 3, we obtain several optimality conditions, as shown next.

Theorem 2 (Sufficient Optimality Conditions and Indexability) *Assume that any of the conditions* (a)-(d) *of Proposition 3 holds. Then $v(\pi)$ solves linear program (P) optimally.*

It is easy to see that conditions (a)-(d) of Proposition 3 are not, in general, necessary optimality conditions. They are neccessary if the polytope $\mathcal{B}(A, b)$ is nondegenerate. Some consequences of Theorem 2 are the following:

Remarks:

1. **Sensitivity analysis:** Optimality condition (c) of Proposition 3 is specially well suited for performing sensitivity analysis. Consider the following question: given a permutation π of E, for what vectors R and matrices A can we guarantee that $v(\pi)$ solves problem (P) optimally? The answer is: for R and A that satisfy the condition

$$R_\pi A_\pi^{-1} \in \Re_-^{n-1} \times \Re.$$

We may also ask: for which permutations π can we guarantee that $v(\pi)$ is optimal? By Proposition 3(d), the answer now is: for permutations π that satisfy

$$\gamma_{\pi_1} \leq \gamma_{\pi_2} \leq \cdots \leq \gamma_{\pi_n},$$

thus providing an $O(n \log n)$ optimality test for π. Glazebrook [17] addressed the problem of sensitivity analysis in stochastic scheduling problems. His results are in the form of suboptimality bounds.

2. **Explicit formulae for Gittins indices:** Proposition 3(c) provides an explicit formula for the vector of generalized Gittins indices. The formula reveals that the indices are piecewise linear functions of the reward vector.

3. **Indexability:** Optimality condition (d) of Proposition 3 shows that any permutation that sorts the generalized Gittins indices in nonincreasing order provides an optimal solution for problem (P). Condition (d) thus shows that this class of optimization problems has an indexability property.

In the case that matrix A has a certain special structure, the computation of the indices of (P) can be simplified. Let E be partitioned as $E = \bigcup_{k=1}^{K} E_k$. For $k = 1, \ldots, K$, let $\mathcal{B}(A^k, b^k)$ be the base of an extended polymatroid; let $x^k = (x_i^k)_{i \in E_k}$; let (P_k) be the following linear program:

$$(P_k) \qquad \max\{ \sum_{i \in E_k} R_i x_i^k : x^k \in \mathcal{B}(A^k, b^k) \}; \tag{16}$$

let $\{\gamma_i^k\}_{i \in E_k}$ be the generalized Gittins indices of problem (P_k). Assume that the following independence condition holds:

$$A_i^S = A_i^{S \cap E_k} = (A^k)_i^{S \cap E_k}, \quad \text{for } i \in S \cap E_k \quad \text{and } S \subseteq E. \tag{17}$$

Under condition (17) there is an easy relation between the indices of problems (P) and (P_k), as shown in the next result.

Theorem 3 (Index Decomposition) *Under condition (17), the generalized Gittins indices of linear programs (P) and (P_k) satisfy*

$$\gamma_i = \gamma_i^k, \quad \textit{for } i \in E_k \quad \textit{and } k = 1, \ldots, K. \tag{18}$$

Proof

Let

$$h_i = \gamma_i^k, \qquad \text{for } i \in E_k \quad \text{and} \quad k = 1, \dots, K. \tag{19}$$

Let us renumber the elements of E so that

$$h_1 \le h_2 \le \cdots \le h_n. \tag{20}$$

Let $\pi = (1, \dots, n)$. Permutation π of E induces permutations π^k of E_k, for $k = 1, \dots, K$, that satisfy

$$\gamma^k_{\pi^k_1} \le \cdots \le \gamma^k_{\pi^k_{|E_k|}}. \tag{21}$$

Hence, by Proposition 3 it follows that

$$\gamma^k_{\pi^k} = R^k_{\pi^k}(A^k_{\pi^k})^{-1}T_k, \qquad \text{for } k = 1, \dots K$$

or, equivalently,

$$(\gamma^1_{\pi_1}, \gamma^2_{\pi_2}, \dots, \gamma^k_{\pi^k}) \begin{pmatrix} T_1^{-1}A^1_{\pi^1} & 0 & \dots & 0 \\ 0 & T_2^{-1}A^2_{\pi^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & T_k^{-1}A^k_{\pi^k} \end{pmatrix} = (R^1_{\pi^1}, R^2_{\pi^2}, \dots, R^k_{\pi^k}). \tag{22}$$

where T_k is an $|E_k| \times |E_k|$ matrix with the same structure as matrix T, for $k = 1, \dots, K$. On the other hand, we have

$$\begin{aligned} T^{-1}A_\pi &= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix} A_\pi \\ &= \begin{pmatrix} A_1^{\{1\}} & 0 & \dots & 0 \\ A_1^{\{1,2\}} - A_1^{\{1\}} & A_2^{\{1,2\}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_1^{\{1,\dots,n\}} - A_1^{\{1,\dots,n-1\}} & A_2^{\{1,\dots,n\}} - A_2^{\{1,\dots,n-1\}} & \dots & A_n^{\{1,\dots,n\}} \end{pmatrix}. \end{aligned}$$

Now, notice that if $i \in E_k$, $j \in E \setminus E_k$ and $i < j$ then, by (17):

$$A_i^{\{1,\dots,j\}} = A_i^{\{1,\dots,j\} \cap E_k} = A_i^{\{1,\dots,j-1\} \cap E_k} = A_i^{\{1,\dots,j-1\}}. \tag{23}$$

Hence, by (19) and (23) it follows that system (22) can be written equivalently as

$$h_\pi T^{-1} A_\pi = R_\pi. \tag{24}$$

Now, (20) and (24) imply that

$$R_\pi A_\pi^{-1} = h_\pi T^{-1} = \begin{pmatrix} h_1 - h_2 \\ h_2 - h_3 \\ \vdots \\ h_{n-1} - h_n \\ h_n \end{pmatrix} \in \Re_-^{n-1} \times \Re, \tag{25}$$

and by Proposition 3 it follows that the generalized Gittins indices of problem (P) satisfy

$$\gamma_\pi = R_\pi A_\pi^{-1} T.$$

Hence, by (24),

$$h_i = \gamma_i, \qquad \text{for } i \in E$$

and this completes the proof of the theorem. □

Theorem 3 implies that the fundamental reason for decomposition to hold is (17). An easy and useful consequence of Theorems 2 and 3 is the following:

Corollary 1 *Under the assumptions of Theorem 3, an optimal solution of problem (P) can be computed by solving the K subproblems (P_k), for $k = 1, \ldots, K$ by algorithm $\mathcal{A}_1$ and computing their respective generalized Gittins indices.*

It is important to emphasize that the index decomposition property is much stronger that the indexability property. We will see later that the classical multi-armed bandit problem has the index decomposition property. On the other hand, we will see that Klimov's problem (see [22]) has the indexability property, but in the general case it is not decomposable.

2.3 Generalized Conservation Laws

Shantikumar and Yao [26] formalized a definition of *strong conservation laws* for performance measures in general multiclass queues, that implies a polymatroidal structure in the performance space. We next present a more general definition of *generalized conservation laws* in a broader context that implies an extended polymatroidal structure in the performance space, which has several interesting and important implications. Consider a general dynamic and stochastic *job scheduling process*. There are n job types, which we label $i \in E = \{1, \ldots, n\}$. We consider the class of *admissible scheduling policies*, which we denote $\mathcal{U}$, to be the class of all nonidling, nonpreemtive and nonanticipative scheduling policies.

Let x_i^u be a performance measure of type i jobs under admissible policy u, for $i \in E$. We assume that x_i^u is an expectation. Let x^u be the corresponding performance vector. Let x^π denote the performance vector under a *fixed priority rule* that assigns priorities to the job types according to the permutation $\pi = (\pi_1, \ldots, \pi_n)$ of E, where type π_n has the highest priority, ..., type π_1 has the lowest priority.

Definition 5 (Generalized Conservation Laws) The performance vector x is said to satisfy *generalized conservation laws* if there exist a function $b : 2^E - \Re_+$ such that $b(\emptyset) = 0$ and a matrix $A = (A_i^S)_{i \in E, S \subseteq E}$ satisfying (1) such that:

(a)

$$b(S) = \sum_{i \in S} A_i^S x_i^\pi, \quad \text{for all } \pi : \{\pi_1, \ldots, \pi_{|S|}\} = S \quad \text{and} \quad S \subseteq E; \tag{26}$$

(b)

$$\sum_{i \in S} A_i^S x_i^u \geq b(S), \quad \text{for all } S \subset E \qquad \text{and} \qquad \sum_{i \in E} A_i^E x_i^u = b(E), \quad \text{for all } u \in \mathcal{U}. \tag{27}$$

In words, a performance vector is said to satisfy generalized conservation laws if: there exist weights A_i^S such that the total weighted performance over all job types is invariant under any admissible policy, and the minimum weighted performance over the job types in any subset $S \subset E$ is achieved by any fixed priority rule that gives priority to all other types (in S^c) over types in S. The strong conservation laws of Shantikumar and Yao [26] correspond to the special case that all weights are $A_i^S = 1$.

The connection between generalized conservation laws and extended polymatroids is the following theorem:

Theorem 4 *Assume that the performance vector x satisfies generalized conservation laws* (26) *and* (27). *Then*

(a) The vertices of $\mathcal{B}(A, b)$ are the performance vectors of the fixed priority rules, and $x^\pi = v(\pi)$, for every permutation π of E.

(b) The extended polymatroid base $\mathcal{B}(A, b)$ is the performance space.

Proof

(a) By (26) it follows that $x^{\pi} = v(\pi)$. And by Theorem 1 the result follows.

(b) Let $X = \{ x^u : u \in \mathcal{U} \}$ be the performance space. Let $\mathcal{B}_v(A,b)$ be the set of extreme points of $\mathcal{B}(A,b)$. By (27) it follows that $X \subseteq \mathcal{B}(A,b)$. By (a), $\mathcal{B}_v(A,b) \subseteq X$. Hence, since X is a convex set ($\mathcal{U}$ contains randomized policies) we have

$$\mathcal{B}(A,b) = \mathrm{conv}(\mathcal{B}(A,b)) \subseteq X.$$

Hence $X = \mathcal{B}(A,b)$, and this completes the proof of the theorem. □

As a consequence of Theorem 4, it follows by Carathéodory theorem that the performance vector x^u corresponding to an admissible policy u can be achieved by a randomization of at most $n+1$ fixed priority rules.

2.4 Optimization over systems satisfying generalized conservation laws

Let x^u be a performance vector for a dynamic and stochastic job scheduling process that satisfies generalized conservation laws (associated with A, $b(\cdot)$). Suppose that we want to find an admissible policy u that maximizes a linear reward function $\sum_{i \in E} R_i x_i^u$. This optimal scheduling control problem can be expressed as

$$(P_{\mathcal{U}}) \qquad \max\{ \sum_{i \in E} R_i x_i^u : u \in \mathcal{U} \}. \tag{28}$$

By Theorem 4 this control problem can be transformed into the following linear programming problem:

$$(P) \qquad \max\{ \sum_{i \in E} R_i x_i : x \in \mathcal{B}(A,b) \}. \tag{29}$$

The strong structural properties of extended polymatroids lead to strong structural properties in the control problem. Suppose that to each job type i we attach an index, γ_i. A policy that selects at each decision epoch a job of currently largest index will be referred to as an ***index policy***.

Let $\gamma_1, \ldots, \gamma_n$ be the generalized Gittins indices of linear program (P). As a direct consequence of the results of Section 2.2 we show next that the control problem $(P_{\mathcal{U}})$ is solved by an index policy, with indices given by $\gamma_1, \ldots, \gamma_n$.

Theorem 5 (Indexability) *(a) Let $v(\pi)$ be an optimal solution of linear program (P). Then the fixed priority rule that assigns priorities to the job types according to permutation*

π is optimal for the control problem $(P_{\mathcal{U}})$;

(b) A policy that selects at each decision epoch a job of currently largest generalized Gittins index is optimal for the control problem.

The previous theorem implies that systems satisfying generalized conservation laws are *indexable systems.*

Let us consider now a dynamic and stochastic *project selection process*, in which there are K project types, labeled $k = 1, \ldots, K$. At each decision epoch a project must be selected. A project of type k can be in one of a finite number of states $i_k \in E_k$. These states correspond to ***stages*** in the development of the project. Clearly this process can be interpreted as a job scheduling process, as follows: simply interpret the action of selecting a project k in state $i_k \in E_k$ as selecting a job of type $i = i_k \in \bigcup_{k'=1}^{K} E_{k'}$. We may interpret that each project consists of several jobs. Let us assume that this job scheduling process satisfies generalized conservation laws associated with matrix A and set function $b(\cdot)$. By Theorem 5, the corresponding optimal control problem is solved by an index policy. We will see next that when a certain independence condition among the projects is satisfied, a strong *index decomposition* property holds.

We thus assume that E is partitioned as $E = \bigcup_{k=1}^{K} E_k$. Let $x^k = (x_i)_{i \in E_k}$ be the performance vector over job types in E_k corresponding to the project selection problem obtained when projects of types other than k are ignored (i.e., they are never engaged). Let us assume that the performance vector x^k satisfies generalized conservation laws associated with matrix A^k and set function $b^k(\cdot)$, and that the independence condition (17) is satisfied. Let $\mathcal{U}_k$ be the corresponding set of admissible policies.

Under these assumptions, Theorem 3 applies, and together with Theorem 5(b) we get the following result:

Theorem 6 (Index Decomposition) *Under condition (17), the generalized Gittins indices of job types in E_k only depend on characteristics of project type k.*

The previous theorem identifies a sufficient condition for the indices of an indexable system to have a strong decomposition property. Therefore, systems that satisfy generalized conservation laws which further satisfy (17) are *decomposable systems*. For such systems the solution of problem $(P_{\mathcal{U}})$ can be obtained by solving K smaller independent subproblems. This theorem justifies the term generalized Gittins indices. We will see in Section 4 that when applied to the multi-armed bandit problem, these indices reduce to the usual Gittins indices.

Let us consider briefly the problem of optimizing a nonlinear cost function on the performance vector. Bhattacharya *et al.* [4] addressed the problems of separable convex, min-max, lexicographic and semi-separable convex optimization over an extended polymatroid, and provided iterative algorithms for their solution. Analogously as what we did in the linear reward case, the control problem in the case of a nonlinear reward function can be reduced to solving a nonlinear programming problem over the base of an extended polymatroid.

3 Branching Bandit Processes

Consider the following *branching bandit process* introduced by Weiss [35], who observed that it can model a large number of dynamic and stochastic scheduling processes. There is a finite number of project types, labeled $k = 1, \ldots, K$. A type k project can be in one of a finite number of states $i_k \in E_k$, which correspond to *stages* in the development of the project. It is convenient in what follows to combine these two indicators into a single label $i = i_k$, the state of a project. Let $E = \cup_{k=1}^{K} E_k = \{1, \ldots, n\}$ be the finite set of possible states of all project types.

We associate with state i of a project a random time v_i and random arrivals $N_i = (N_{ij})_{j \in E}$. Engaging the project keeps the system busy for a duration v_i (the duration of stage i), and upon completion of the stage the project is replaced by a nonnegative integer number of new projects N_{ij}, in states $j \in E$. We assume that given i, the durations and the descendants v_i, N_i are random variables with an arbitrary joint distribution, independent of all other projects, and identically distributed for the same i. Projects are to be selected under a nonidling, nonpreemptive and nonanticipative scheduling policy u. We shall refer to this class of policies, which we denote $\mathcal{U}$, as the class of *admissible policies*. The decision epochs are $t = 0$ and the instants at which a project stage is completed and there is some project present. If m_i is the number of projects in state i present at a given time, then it is clear that this process is a semi-Markov decision process with states $m = (m_1, \ldots, m_n)$.

The model of arm-acquiring bandits (see Whittle [37], [38]) is a special case of branching bandit process, in which the descendants N_i consist of two parts: (1) a transition of the project engaged to a new state, and (2) external arrivals of new projects, independent of i

or of the transition. The classical multi-armed bandit problem corresponds to the special case that there are no external arrivals of projects, and the stage durations are 1.

The branching bandit process is thus a special case of a project selection process. Therefore, as described in Subsection 2.4, it can be interpreted as a job scheduling process. Engaging a type i job in the job scheduling model corresponds to selecting a project of state i in the branching bandit model. We may interpret that each project consists of several jobs. In the analysis that follows, we shall refer to a project in state i as a *type i job*. In this section, we will define two different performance measures for a branching bandit process. The first one will be appropriate for modelling a discounted reward-tax structure. The second one will allow us to model an undiscounted tax structure. In each case we will show that they satisfy generalized conservation laws, and that the corresponding optimal control problem can be solved by a direct application of the results of Section 2.

Let $S \subseteq E$ be a subset of job types. We shall refer to jobs with types in S as S-jobs. Assume now that at time $t = 0$ there is only a single job in the system, which is of type i. Consider the sequence of successive job selections corresponding to an admissible policy u that gives complete priority to S-jobs. This sequence proceeds until all S-jobs are exhausted for the first time, or indefinitely. Call this an (i, S) *period*. Let T_i^S be the duration (possibly infinite) of an (i, S) period. It is easy to see that the distribution of T_i^S is independent of the admissible policy used, as long as it gives complete priority to S-jobs. Note that an $(i, \emptyset)$ period is distributed as v_i. It will be convenient to introduce the following additional notation:

$v_{i,k}$ = duration of the kth selection of a type i job; notice that the distribution of $v_{i,k}$ is independent of k (v_i).

$\tau_{i,k}$ = time at which the kth selection of a type i job occurs;

ν_i = number of times a type i job is selected (can be infinity);

$\{T_{i,k}^S\}_{k \geq 1}$= duration of the (i, S)-period that starts with the kth selection of a type i job. type i job for the kth time.

$Q_i(t)$ = number of type i jobs in the system at time t. $Q(t)$ denotes the vector of the $Q_i(t)$'s. We assume $Q(0) = (m_1, \ldots, m_n)$ is known.

T_m^S = time until all S-jobs are exhausted for the first time (can be infinity); note that T_m^E is the duration of the busy period.

$$I_i(t) = \begin{cases} 1, & \text{if a type } i \text{ job is being engaged at time } t; \\ 0, & \text{otherwise,} \end{cases}$$

$\Delta_{i,k}^S = \inf\{\, \Delta \geq v_{i,k} : \sum_{j \in S} I_j(\tau_{i,k} + \Delta) = 1 \,\}$, for $i \in S$; note that $\Delta_{i,k}^S$ is the interval between the kth selection of a type i job and the next selection, if any, of an S-job. If no more jobs in S are selected, then $\Delta_{i,k}^S = T_{i,k}^{S^c}$, the remaining interval of the busy period.

$\Delta_m^S = \inf\{\, t : \sum_{i \in S} I_i(t) = 1 \,\}$; note that Δ_m^S is the interval until the first job in S, if any, is selected. If no job in S is selected, $\Delta_m^S = T_m^E$, the busy period.

Proposition 4 *Assume that jobs are selected in the branching bandit process under an admissible policy. Then, for every $S \subseteq E$:*

(a) *If the policy gives complete priority to S^c-jobs then the busy period $[0, T_m^E)$ can be partitioned as follows:*

$$[0, T_m^E) = [0, T_m^{S^c}) \bigcup_{i \in S} \bigcup_{k=1}^{\nu_i} [\tau_{i,k}, \tau_{i,k} + T_{i,k}^{S^c}) \qquad w.\ p.\ 1. \tag{30}$$

(b) *The busy period $[0, T_m^E)$ can be partitioned as follows:*

$$[0, T_m^E) = [0, \Delta_m^S) \bigcup_{i \in S} \bigcup_{k=1}^{\nu_i} [\tau_{i,k}, \tau_{i,k} + \Delta_{i,k}^S) \qquad w.\ p.\ 1. \tag{31}$$

(c) *The following inequalities hold w. p. 1:*

$$\Delta_{i,k}^S \leq T_{i,k}^{S^c}, \tag{32}$$

and

$$\Delta_m^S \leq T_m^{S^c}. \tag{33}$$

Proof

(a) Intuitively (30) expresses the fact that under a policy that gives complete priority to S^c-jobs, the duration of a busy period is partitioned into (1) the initial interval in which all jobs in S^c are exhausted for the first time, and (2) intervals in which all jobs in S are exhausted, given that after working on a job in S we clear first all jobs in S^c that were generated.

More formally, let u be an admissible policy that gives complete priority to S^c-jobs. It is easy to see then that the intervals in the right hand side of (30) are disjoint. Moreover, the inclusion $\supseteq$ is obvious. In order to show that (30) is indeed a partition, let us show the inclusion $\subseteq$. Let $t \in [0, T_m^E) \setminus [0, T_m^{S^c})$, otherwise we are done. Since u is a nonidling policy, at time t some job is being engaged. Let j be the type of this job. If $j \in S$ then it is clear that $t \in [\tau_{j,k}, \tau_{j,k} + T_{j,k}^{S^c})$ for some k, and we are done. Let us assume that $j \in S^c$. Let us define

$$D = \{\tau_{i,k} : i \in S, k \in \{1, \ldots, \nu_i\}, \text{and } \tau_{i,k} \leq t\}.$$

Since $t > T_m^{S^c} \in D$ it follows that $D \neq \emptyset$. Now, since by hypothesis $\mathrm{E}[v_i] > 0$, for all i, it follows that D is a finite set. Let $i^* \in S$ and k^* be such that

$$\tau_{i^*,k^*} = \max_{\tau \in D} \tau.$$

Assume that $\tau_{i^*,k^*} + T_{i^*,k^*}^{S^c} \leq t$. Now, $\tau_{i^*,k^*} + T_{i^*,k^*}^{S^c}$ is a decision epoch at which S^c is empty. Since the policy is nonidling, it follows that at this epoch one starts working on some type i job, with $i \in S$, that is, $\tau_{i,k} = \tau_{i^*,k^*} + T_{i^*,k^*}^{S^c}$, contradicting the definition of τ_{i^*,k^*}. Hence, it must be $t < \tau_{i^*,k^*} + T_{i^*,k^*}^{S^c}$. And by definition of D it follows that $t \in [\tau_{i*,k*}, \tau_{i*,k*} + T_{i*,k*}^{S^c})$, and this completes the proof of the proposition.

(b) Equality (31) formalizes the fact that under an admissible policy the busy period can be decomposed into (1) the interval until the first job in S is selected, (2) the disjoint union of the intervals between selections of successive S-jobs and (3) the interval between the last selection of a job in S and the end of the busy period. Note that if no S-job is selected, then $\nu_i = 0$, for $i \in S$, and $\Delta_m^S = T_m^E$, thus reducing the partition to a single interval.

(c) Let $\tau_{i,k}$ be the time of the kth selection of a type i job ($i \in S$). Since the next selection (if any) of an S-job can occur, at most, at the end of the (i, S^c) period $[\tau_{i,k}, \tau_{i,k} + T_i^{S^c})$, inequality (32) follows. On the other hand, since the time until the first selection of an S-job, Δ_m^S, can be at most the duration of the initial (i, S^c) period, (33) follows. □

3.1 Discounted Branching Bandits

In this subsection we will introduce a family of performance measures for branching bandits, $\{x^u(\alpha)\}_{\alpha>0}$, that satisfy generalized conservation laws. They are appropriate for modelling a linear discounted reward-tax structure on the branching bandit process. We have already defined the indicator

$$I_i(t) = \begin{cases} 1, & \text{if a type } i \text{ job is being engaged at time } t; \\ 0, & \text{otherwise,} \end{cases} \tag{34}$$

and, for a given $\alpha > 0$, we define

$$x_i^u(\alpha) = \mathrm{E}_u\left[\int_0^\infty e^{-\alpha t} I_i(t)\, dt\right] = \int_0^\infty \mathrm{E}_u[I_i(t)]\, e^{-\alpha t}\, dt, \quad i \in E. \tag{35}$$

3.1.1 Generalized Conservation Laws

In this section we prove that the performance measure for branching bandits defined in (35) satisfies generalized conservation laws. Let us define

$$A_{i,\alpha}^S = \frac{\mathrm{E}[\int_0^{T_i^{S^c}} e^{-\alpha t}\, dt]}{\mathrm{E}[\int_0^{v_i} e^{-\alpha t}\, dt]}, \quad i \in S \tag{36}$$

and

$$b_\alpha(S) = \mathrm{E}\left[\int_0^{T_m^E} e^{-\alpha t}\, dt\right] - \mathrm{E}\left[\int_0^{T_m^{S^c}} e^{-\alpha t}\, dt\right]. \tag{37}$$

The main result is the following

Theorem 7 (Generalized Conservation Laws for Discounted Branching Bandits) *The performance vector for branching bandits $x^u(\alpha)$ satisfies generalized conservation laws* (26) *and* (27) *associated with matrix A_α and set function $b_\alpha(\cdot)$.*

Proof

Let $S \subseteq E$. Let us assume that jobs are selected under an admissible policy u. This generates a branching bandit process. Let us define two random vectors, $(r_i^{\mathrm{I}})_{i\in E}$ and $(r_i^{\mathrm{II},S})_{i\in S}$, as functions of its sample path as follows:

$$\begin{aligned} r_i^{\mathrm{I}} &= \int_0^\infty I_i(t) e^{-\alpha t}\, dt = \sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+v_{i,k}} e^{-\alpha t}\, dt \\ &= \sum_{k=1}^{\nu_i} e^{-\alpha \tau_{i,k}} \int_0^{v_{i,k}} e^{-\alpha t}\, dt, \end{aligned} \tag{38}$$

and

$$r_i^{\mathrm{II},S} = \sum_{k=1}^{\nu_i} e^{-\alpha \tau_{i,k}} \int_0^{T_{i,k}^{S^c}} e^{-\alpha t}\, dt, \quad i \in S. \tag{39}$$

Now, we have

$$
\begin{aligned}
x_i^u(\alpha) &= \mathrm{E}_u[r_i^{\mathrm{I}}] = \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}} \int_0^{v_{i,k}} e^{-\alpha t}\, dt\right] \\
&= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \mathrm{E}\left[e^{-\alpha\tau_{i,k}} \int_0^{v_{i,k}} e^{-\alpha t}\, dt \mid \nu_i\right]\right] \\
&= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \mathrm{E}[e^{-\alpha\tau_{i,k}} \mid \nu_i\,]\, \mathrm{E}\left[\int_0^{v_{i,k}} e^{-\alpha t}\, dt\right]\right] \qquad (40) \\
&= \mathrm{E}\left[\int_0^{v_i} e^{-\alpha t}\, dt\right] \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}}\right] \qquad (41)
\end{aligned}
$$

Note that equality (40) holds because, since u is nonanticipative, $\tau_{i,k}$ and $v_{i,k}$ are independent random variables. On the other hand, we have

$$
\begin{aligned}
\mathrm{E}_u[r_i^{\mathrm{II},S}] &= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}} \int_0^{T_{i,k}^{S^c}} e^{-\alpha t}\, dt\right] = \mathrm{E}_u\left[\mathrm{E}\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}} \int_0^{T_{i,k}^{S^c}} e^{-\alpha t}\, dt \mid \nu_i\right]\right] \\
&= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \mathrm{E}\left[\int_0^{T_i^{S^c}} e^{-\alpha t}\, dt\right] \mathrm{E}\left[e^{-\alpha\tau_{i,k}} \mid \nu_i\right]\right] \qquad (42) \\
&= \mathrm{E}\left[\int_0^{T_i^{S^c}} e^{-\alpha t}\, dt\right] \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}}\right] \\
&= A_{i,\alpha}^S\, \mathrm{E}\left[\int_0^{v_i} e^{-\alpha t}\, dt\right] \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}}\right]. \qquad (43)
\end{aligned}
$$

Note that equality (42) holds because, since u is nonanticipative, $\tau_{i,k}$ and $T_{i,k}^{S^c}$ are independent. Hence, by (41) and (43)

$$
\mathrm{E}_u[r_i^{\mathrm{II},S}] = A_{i,\alpha}^S x_i^u(\alpha), \quad i \in S, \qquad (44)
$$

and we obtain:

$$
\mathrm{E}_u\left[\sum_{i\in S} r_i^{\mathrm{II},S}\right] = \sum_{i\in S} A_{i,\alpha}^S x_i^u(\alpha). \qquad (45)
$$

We first show that generalized conservation law (26) holds. Consider a policy π that gives complete priority to S^c-jobs. Applying Proposition 4 (part (a)), we obtain:

$$
\begin{aligned}
\int_0^{T_m^E} e^{-\alpha t}\, dt &= \int_0^{T_m^{S^c}} e^{-\alpha t}\, dt + \sum_{i\in S} \sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} e^{-\alpha t}\, dt \\
&= \int_0^{T_m^{S^c}} e^{-\alpha t}\, dt + \sum_{i\in S} \sum_{k=1}^{\nu_i} e^{-\alpha\tau_{i,k}} \int_0^{T_{i,k}^{S^c}} e^{-\alpha t}\, dt \\
&= \int_0^{T_m^{S^c}} e^{-\alpha t}\, dt + \sum_{i\in S} r_i^{\mathrm{II},S}. \qquad (46)
\end{aligned}
$$

Hence, taking expectations and using equation (45) we obtain

$$\mathrm{E}\left[\int_0^{T_m^E} e^{-\alpha t}\,dt\right] = \mathrm{E}\left[\int_0^{T_m^{S^c}} e^{-\alpha t}\,dt\right] + \sum_{i\in S} A_{i,\alpha}^S x_i^\pi(\alpha)$$

or equivalently, by (37),

$$\sum_{i\in S} A_{i,\alpha}^S x_i^\pi(\alpha) = b_\alpha(S),$$

which proves that generalized conservation law (26) holds.

We next show that generalized conservation law (27) is satisfied. Since jobs are selected under admissible policy u, Proposition 4 (part (b)) applies, and we can write

$$\int_0^{T_m^E} e^{-\alpha t}\,dt = \int_0^{\Delta_m^S} e^{-\alpha t}\,dt + \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+\Delta_{i,k}^S} e^{-\alpha t}\,dt. \qquad (47)$$

On the other hand, we have

$$\begin{aligned}
\sum_{i\in S} r_i^{\mathrm{II},S} &= \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} e^{-\alpha t}\,dt \\
&\geq \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+\Delta_{i,k}^S} e^{-\alpha t}\,dt && (48)\\
&= \int_0^{T_m^E} e^{-\alpha t}\,dt - \int_0^{\Delta_m^S} e^{-\alpha t}\,dt && (49)\\
&\geq \int_0^{T_m^E} e^{-\alpha t}\,dt - \int_0^{T_m^{S^c}} e^{-\alpha t}\,dt && (50)\\
&= \int_0^{T_m^E} e^{-\alpha t}\,dt - \int_0^{T_m^{S^c}} e^{-\alpha t}\,dt. && (51)
\end{aligned}$$

Notice that (48) follows by Proposition 4 (part (c)), (49) follows by (47), and (50) by Proposition 4 (part (c)). Hence, taking expectations in (51), and applying (45) we obtain

$$\begin{aligned}
\sum_{i\in S} A_{i,\alpha}^S x_i^u(\alpha) &= \mathrm{E}_u[\sum_{i\in S} r_i^{II,S}] \\
&\geq \mathrm{E}\left[\int_0^{T_m^E} e^{-\alpha t}\,dt\right] - \mathrm{E}\left[\int_0^{T_m^{S^c}} e^{-\alpha t}\,dt\right] \\
&= b_\alpha(S) && (52)
\end{aligned}$$

which proves that generalized conservation law (27) holds, and this completes the proof of the theorem. □

Hence, by the results of Subsection 2.3 we obtain:

Corollary 2 *The performance space for branching bandits corresponding to the performance vector $x^u(\alpha)$ is the extended polymatroid base $\mathcal{B}(A_\alpha, b_\alpha)$; furthermore, the vertices of $\mathcal{B}(A_\alpha, b_\alpha)$ are the performance vectors corresponding to the fixed priority rules.*

3.1.2 The Discounted Reward-Tax Problem

Let us associate with a branching bandit process the following linear reward-tax structure: An instantaneous reward of R_i is received at the completion epoch of a type i job. In addition, a holding tax C_i is incurred continuously during the interval that a type i job is in the system. Rewards and taxes are discounted in time with a discount factor $\alpha > 0$. Let us denote

$V_{u,\alpha}^{(R,C)}(m) =$ expected total present value of rewards received minus taxes incurred under policy u, given that there are initially m_i jobs of type i in the system, for $i \in E$.

The discounted reward-tax problem is the following optimal control problem: find an admissible policy u^* that maximizes $V_{u,\alpha}^{(R,C)}(m)$ over all admissible policies u. In this section we reduce the reward-tax problem to the pure rewards case (where $C = 0$). We also find a closed formula for $V_{u,\alpha}^{(R,C)}(m)$ and show how to solve the problem using algorithm $\mathcal{A}_1$.

The Pure Rewards Case.

Let us introduce the transform of v_i, i.e., $\Psi_i(\theta) = \mathrm{E}[e^{-\theta v_i}]$. We then have

$$\begin{aligned}
V_{u,\alpha}^{(R,0)}(m) &= \mathrm{E}_u\left[\sum_{i\in E}\sum_{k=1}^{\nu_i} R_i e^{-\alpha(\tau_{i,k}+v_{i,k})}\right] \\
&= \sum_{i\in E} R_i \mathrm{E}[e^{-\alpha v_i}]\, \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} e^{-\alpha \tau_{i,k}}\right] \qquad (53)\\
&= \sum_{i\in E} \frac{\mathrm{E}[e^{-\alpha v_i}]}{\mathrm{E}[\int_0^{v_i} e^{-\alpha t}\,dt]} R_i x_i^u(\alpha) \qquad (54)\\
&= \sum_{i\in E} \frac{\alpha \Psi_i(\alpha)}{1-\Psi_i(\alpha)} R_i x_i^u(\alpha). \qquad (55)
\end{aligned}$$

Notice that equality (54) holds by (41).

It is also straightforward to model the case in which rewards are received continuously during the interval that a type i job is in the system rather than at a completion epoch. Let $\hat{V}_{u,\alpha}^{(R,0)}(m)$ be the expected total present value of rewards. Then

$$\hat{V}_{u,\alpha}^{(R,0)}(m) = \mathrm{E}_u\left[\sum_{i\in E} R_i \int_0^\infty e^{-\alpha t} I_i(t)\,dt\right] = \sum_{i\in E} R_i x_i^u(\alpha).$$

The Reward-Tax Problem; Reduction to the Pure Rewards Case

We will next show how to reduce the reward-tax problem to the pure rewards case using the following idea introduced by Bell [1] (see also Harrison [18], Stidham [28] and Whittle [38] for further discussion). The expected present value of holding taxes is the same whether they are charged continuously in time, or according to the following charging scheme: At the arrival epoch of a type *i* job, charge the system with an instantaneous *entrance charge* of (C_i/α), equal to the total discounted continuous holding cost that would be incurred if the job remained within the system forever; at the departure epoch of the job (if it ever departs), credit the system with an instantaneous ***departure refund*** of (C_i/α), thus refunding that portion of the entrance cost corresponding to residence beyond the departure epoch. Therefore, we can write

$$\begin{aligned}
V_{u,\alpha}^{(R,C)}(m) &= \mathrm{E}_u[\,\text{Rewards}\,] - \mathrm{E}_u[\,\text{Charges at } t = 0\,] + \\
&\quad (\,\mathrm{E}_u[\,\text{Departure refunds}\,] - \mathrm{E}_u[\,\text{Entrance Charges}\,]\,) \\
&= V_{u,\alpha}^{(R,0)}(m) - \sum_{i \in E} m_i(C_i/\alpha) + V_{u,\alpha}^{(R',0)}(m) \\
&= V_{u,\alpha}^{(R+R',0)}(m) - \sum_{i \in E} m_i(C_i/\alpha) \\
&= \sum_{i \in E} \Big\{ R_i + \frac{C_i - \sum_{j \in E} \mathrm{E}[N_{ij}] C_j}{\alpha} \Big\} \frac{\alpha \Psi_i(\alpha)}{1 - \Psi_i(\alpha)} x_i^u(\alpha) - \sum_{i \in E} m_i(C_i/\alpha). \qquad (56)
\end{aligned}$$

where

$$R_i' = (C_i/\alpha) - \sum_{j \in E} \mathrm{E}[\,N_{ij}\,](C_j/\alpha). \qquad (57)$$

From equation (56) it is straightforward to apply the results of Section 2 to solve the control problem: use algorithm $\mathcal{A}_1$ with input $(\hat{R}_\alpha, A_\alpha)$, where

$$\hat{R}_{i,\alpha} = \Big\{ R_i + \frac{C_i - \sum_{j \in E} \mathrm{E}[N_{ij}]\, C_j}{\alpha} \Big\} \frac{\alpha \Psi(\alpha)}{1 - \Psi(\alpha)}. \qquad (58)$$

Let $\gamma_1(\alpha), \ldots, \gamma_n(\alpha)$ be the corresponding generalized Gittins indices. Then we have

Theorem 8 (Optimality and Indexability: Discounted Branching Bandits) (a) *Algorithm $\mathcal{A}_1$ provides an optimal policy for the discounted reward-tax branching bandit problem;*

(b) *An optimal policy is to work at each decision epoch on a project with largest index $\gamma_i(\alpha)$.*

The previous theorem characterizes the structure of the optimal policy. Moreover, since in Proposition 6 below, we find closed form expressions for the matrix A_α and the set function $b_\alpha(\cdot)$, we can compute not only the structure, but also the performance of the optimal policy (optimal profit, optimal extreme point of the extended polymatroid). Note also that the decomposition of the indices does not hold in the general case: in other words, the generalized indices of the states of a type k project depend in general on characteristics of project types other than k, i.e., branching bandits is an example of an indexable but not decomposable system. We may also prove the following result:

Theorem 9 (Continuity of generalized Gittins indices) *The generalized Gittins indices $\gamma_1(\alpha), \ldots, \gamma_n(\alpha)$ are continuous functions of the discount factor α, for $\alpha > 0$.*

Proof

It is easy to see that the generalized Gittins indices depend continuously on the input of algorithm $\mathcal{A}_1$. Also, since the function $\alpha \mapsto (\hat{R}_\alpha, A_\alpha)$ is continuous the result follows. □

The Pure Tax Case: Minimizing Time-Dependent Expected Number in System

In several applications of branching bandits (for example queueing systems) one is often interested in minimizing a weighted sum of discounted time-dependent expected number of jobs in the system. Let $Q_j^{*u}(\cdot)$ denote the Laplace transform of the time-dependent expected number of type j jobs in the system under policy u, i.e.,

$$Q_j^{*u}(\theta) = \int_0^\infty \mathrm{E}_u[\,Q_j(t)|Q(0) = m\,]\,e^{-\theta t}\,dt, \quad j \in E. \tag{59}$$

An interesting optimization problem is to:

$$\min_{u \in U} \sum_{j \in E} C_j Q_j^{*u}(\alpha).$$

The problem can be modelled as a pure tax problem as follows:

$$\sum_{j \in E} C_j Q_j^{*u}(\alpha) = -V_{u,\alpha}^{(0,C)}(m).$$

and thus by making $R = 0$, $C_j = 1$ and $C_i = 0$ for $i \neq j$ in (56) we obtain

$$Q_j^{*u}(\alpha) = \frac{m_j}{\alpha} - \frac{\Psi_j(\alpha)}{1 - \Psi_j(\alpha)} x_j^u(\alpha) + \sum_{i \in E} \frac{\Psi_i(\alpha)}{1 - \Psi_i(\alpha)} \mathrm{E}[\,N_{ij}\,] x_i^u(\alpha), \quad j \in E. \tag{60}$$

See Harrison [18] for a similar result in the context of a multiclass queue.

3.1.3 Interpretation of Generalized Gittins Indices in Discounted Branching Bandits

Consider the following modification of the branching bandits problem: We modify the original problem by adding an additional project type, which we call 0, with only one state/stage of infinite duration, that is, $v_0 = \infty$ with probability 1. A reward of R_0, continuously discounted, is received for each unit of time that a type 0 project is engaged. Notice that the choice of working on project type 0 can be interpreted as the choice of retirement from the original problem for a *pension* of R_0, continuously discounted in time.

Now, the modified problem is still a branching bandits problem. Let us assume that at time $t = 0$ there are only two projects present, one of type 0 and another in state $i \in E$. We may then ask the following question: Which is the smallest value of the pension R_0 which makes the option of retirement (working on project type 0) preferable to the option of continuation (working on the project in state i)? Let us call this *equitable surrender value* $R_0^*(i)$. We have then the following result:

Proposition 5 *The generalized Gittins index of project state i in the original branching bandits problem coincides with the equitable surrender value of state i, $R_0^*(i)$.*

Proof

Let $\gamma_1, \ldots, \gamma_n$ be the generalized Gittins indices corresponding to the original branching bandits problem. Let $\gamma_0^0, \gamma_1^0, \ldots, \gamma_n^0$ be the generalized Gittins indices for the modified problem. Let us partition the modified state space as $\hat{E} = \{0\} \cup E$. It is easy to verify that the decomposition condition (17) holds. Hence Theorem 3 applies, and therefore we have

$$\gamma_0^0 = R_0 \qquad \text{and} \qquad \gamma_j^0 = \gamma_j, \quad j \in E. \tag{61}$$

Now, since by Theorem 8 it is optimal to work on a project with largest current generalized Gittins index, it follows that the surrender reward R_0 which makes the options of continuation and of retirement (with reward R_0) equally attractive is $R_0 = \gamma_0^0$. But by definition $R_0^*(i)$ is such a breakpoint. Therefore $R_0^*(i) = \gamma_0^0$, and the proof is complete. □

Whittle [36], [38] introduced the idea of a retirement option in his analysis of the multi-armed bandit problem, and provided an interpretation of the Gittins indices as equitable surrender values. Weber [34] also makes use of this characterization of the Gittins indices in his intuitive proof. Here we extend this interpretation to the more general case of branching bandits. From this characterization it follows that the generalized Gittins indices coincide indeed with the well known Gittins indices in the classical multi-armed bandit problem, which justifies their name.

3.1.4 Computation of A_α and $b_\alpha(\cdot)$

The results of the previous sections are structural, but do not lead to explicit computations of the matrix A_α and the set function $b_\alpha(\cdot)$ appearing in the generalized conservation laws (26) and (27) for the branching bandit problem. Our goal in this section is to compute from generic data the matrix A_α and the set function $b_\alpha(\cdot)$. Combined with the previous results these computations make it possible to evaluate the performance of specific policies as well as the optimal policy.

As generic data for the branching bandit process, we assume that the joint distribution of $v_i, (N_{ij})_{j\in E}$ is given by the transform

$$\Phi_i(\theta, z_1, \ldots, z_n) = \mathrm{E}\left[e^{-\theta v_i} z_1^{N_{i1}} \ldots z_n^{N_{in}} \right]. \tag{62}$$

In addition, we have already introduced the the generating function of the marginal distribution of v_i (denoted $G_i(\cdot)$):

$$\Psi_i(\theta) = \mathrm{E}[e^{-\theta v_i}] = \int_0^\infty e^{-\theta t}\, dG_i(t). \tag{63}$$

Finally the vector $m = (m_1, \ldots, m_n)$ of jobs initially present is given.

As we saw in the previous section the duration of an (i, S)-period, T_i^S, plays a crucial role. We will compute its moment generating function

$$\Psi_i^S(\theta) = \mathrm{E}[e^{-\theta T_i^S}]. \tag{64}$$

For this reason we decompose the duration of an (i, S)-period as a sum of independent random variables as follows:

$$T_i^S \overset{\mathrm{d}}{=} v_i + \sum_{j\in S} \sum_{k=1}^{N_{i,j}} T_{j,k}^S, \tag{65}$$

where v_i, $\{T_{j,k}^S\}_{k\geq 1}$ are independent. Therefore,

$$\begin{aligned} \Psi_i^S(\theta) &= \mathrm{E}\left[e^{-\theta v_i}\, \mathrm{E}\left[e^{-\theta \sum_{j\in S} \sum_{k=1}^{N_{ij}} T_{j,k}^S} \mid v_i \right] \right] \\ &= \mathrm{E}\left[e^{-\theta v_i} \prod_{j\in S} \mathrm{E}[e^{-\theta T_j^S}]^{N_{ij}} \right] \\ &= \Phi_i\left(\theta, (\Psi_j^S(\theta))_{j\in S}, 1_{S^c}\right), \quad i\in E. \end{aligned} \tag{66}$$

Given S, fixed point system (66) provides a way to compute the values of $\Psi_i^S(\theta)$, for $i \in E$.

We now have the elements to prove the following result:

Proposition 6 (Computation of A_α and $b_\alpha(\cdot)$) *For a branching bandit process, matrix A_α and set function $b_\alpha(\cdot)$ satisfy the following relations:*

$$A_{i,\alpha}^S = \frac{1 - \Psi_i^{S^c}(\alpha)}{1 - \Psi_i(\alpha)}, \quad i \in S, \quad S \subseteq E; \tag{67}$$

$$b_\alpha(S) = \frac{1}{\alpha} \prod_{j \in S^c} [\Psi_j^{S^c}(\alpha)]^{m_j} - \frac{1}{\alpha} \prod_{j \in E} [\Psi_j^E(\alpha)]^{m_j}, \quad S \subseteq E \tag{68}$$

Proof

Relation (67) follows directly from the definition of $A_{i,\alpha}^S$. On the other hand, we have

$$T_m^S \overset{\mathrm{d}}{=} \sum_{i \in S} \sum_{k=1}^{m_i} T_{i,k}^S. \tag{69}$$

Hence,

$$\begin{aligned} \mathrm{E}\Big[\int_0^{T_m^S} e^{-\alpha t}\, dt\Big] &= \frac{1}{\alpha} - \frac{1}{\alpha}\mathrm{E}\Big[e^{-\alpha \sum_{i \in S} \sum_{k=1}^{m_i} T_{i,k}^S}\Big] \\ &= \frac{1}{\alpha} - \frac{1}{\alpha} \prod_{i \in S} [\Psi_i^S(\alpha)]^{m_i}. \end{aligned} \tag{70}$$

Therefore, from (37), (68) follows. □

Remarks:

1. Note that $A_{i,\alpha}^E = 1$, for $i \in E$, and $b_\alpha(E) = \frac{1}{\alpha} - \frac{1}{\alpha}\prod_{j \in E}[\Psi_j^E(\alpha)]^{m_j}$, $S \subseteq E$.

2. From Proposition 6 we can compute matrix A_α and set function $b_\alpha(\cdot)$ provided we can solve system (66). As an example, we illustrate the form of the equations in the special case, in which the type j jobs that arrive during the time that we work on type i job form a Poisson process with rate λ_{ij}, i.e.,

$$\mathbf{\Phi}_i(\alpha, z_1, \ldots, z_n) = \mathrm{E}\Big[e^{-v_i(\alpha + \sum_{j \in E} \lambda_{ij}(1 - z_j))}\Big] = \Psi_i(\alpha + \sum_{j \in E} \lambda_{ij}(1 - z_j)).$$

In this case, (66) becomes

$$\Psi_i^S(\alpha) = \Psi_i(\alpha + \sum_{j \in S} \lambda_{ij}[1 - \Psi_j^S(\alpha)]), i \in E \tag{71}$$

As a result, an algorithm to compute $\Psi_i^S(\alpha)$ is as follows:

(1) Find a fixed point for the system of nonlinear equations (71) in terms of $\Psi_i^S(\alpha)$. Although in general (71) might not have a closed form solution, in special cases (v_i exponential) a closed form solution could be obtained.

(2) From Proposition 6 compute (A_α, b_α) in terms of $\Psi_i^{S^c}(\alpha)$.

3.2 Undiscounted Branching Bandits

In this section we address branching bandits with no discounts. Clearly, in the case of pure rewards the problem is trivial, since all policies have the same reward. Under a linear undiscounted tax structure on the branching bandit process, however, the problem becomes interesting. Indeed, since an optimal policy under the time average holding cost criterion, also minimizes the expected total holding cost in each busy period (see Nain *et al.* [24]), modelling and solving undiscounted branching bandits leads to the solution of several classical queueing scheduling problems.

More importantly, our approach reveals rigorously the connections of discounted and undiscounted problems, which, in our opinion, has not been thouroughly addressed in the literature. To give a concrete example: after solving an indexable discounted scheduling problem, researchers say that the same ordering of the jobs holds for the undiscounted problem as the discount factor $\alpha \longrightarrow 0$, provided there are no ties of the corresponding indices. It is not clear, however, what happens when there are ties.

We will introduce in this subsection a performance measure z^u for a branching bandit process that satisfies generalized conservation laws. It is appropriate for modelling a linear undiscounted tax structure on the branching bandit process. We shall assume in the following development that all the expectations that appear are finite. We will show later necessary and sufficient conditions for this assumption to hold. Using the indicator

$$I_i(t) = \begin{cases} 1, & \text{if a type } i \text{ job is being engaged at time } t; \\ 0, & \text{otherwise,} \end{cases}$$

we introduced earlier, we let

$$z_i^u = \mathrm{E}_u\left[\int_0^\infty I_i(t)t\,dt\right], \quad i \in E. \tag{72}$$

Let us define

$$A_i^S = \frac{\mathrm{E}[T_i^{S^c}]}{\mathrm{E}[v_i]}, \quad i \in S, \tag{73}$$

and

$$b(S) = \frac{1}{2}\mathrm{E}[(T_m^E)^2] - \frac{1}{2}\mathrm{E}[(T_m^{S^c})^2] + \sum_{i\in S} b_i(S), \tag{74}$$

where

$$b_i(S) = \frac{\mathrm{E}[\nu_i]\,\mathrm{E}[v_i^2]}{2}\left(\frac{\mathrm{E}[T_i^{S^c}]}{\mathrm{E}[v_i]} - \frac{\mathrm{E}[(T_i^{S^c})^2]}{\mathrm{E}[v_i^2]}\right), \quad i \in S. \tag{75}$$

3.2.1 Generalized Conservation Laws

We prove next that the performance measure for a branching bandit process defined in (72) satisfies generalized conservation laws. The main result is the following:

Theorem 10 (Generalized Conservation Laws for Undiscounted Branching Bandits) *The performance vector for branching bandits z^u satisfies generalized conservation laws* (26) *and* (27) *associated with matrix A and set function $b(\cdot)$.*

Proof

Let $S \subseteq E$. Let us assume that jobs are selected under an admissible policy u. This generates a branching bandit process. Let us define two random vectors, $(r_i^{\mathrm{I}})_{i\in E}$ and $(r_i^{\mathrm{II},S})_{i\in S}$, as functions of the sample path as follows:

$$\begin{aligned} r_i^{\mathrm{I}} &= \int_0^\infty I_i(t) t\, dt = \sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+v_{i,k}} t\, dt \\ &= \sum_{k=1}^{\nu_i} (v_{i,k}\, \tau_{i,k} + \frac{v_{i,k}^2}{2}), \quad i \in E, \end{aligned} \tag{76}$$

and

$$r_i^{\mathrm{II},S} = \sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} t\, dt, \quad i \in S. \tag{77}$$

Now, we have

$$\begin{aligned} z_i^u &= \mathrm{E}_u[r_i^{\mathrm{I}}] = \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \mathrm{E}\left[(v_{i,k}\, \tau_{i,k} + \frac{v_{i,k}^2}{2})\,|\nu_i\right]\right] \\ &= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i}\left(\mathrm{E}[v_i]\, \mathrm{E}[\tau_{i,k}|\nu_i] + \frac{\mathrm{E}[v_i^2]}{2}\right)\right] && (78) \\ &= \mathrm{E}[v_i]\, \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \tau_{i,k}\right] + \frac{\mathrm{E}[\nu_i]\, \mathrm{E}[v_i^2]}{2}. && (79) \end{aligned}$$

Note that equality (78) holds because, since u is nonanticipative, $\tau_{i,k}$ and $v_{i,k}$ are independent random variables. On the other hand, we have

$$\begin{aligned} \mathrm{E}_u[r_i^{\mathrm{II},S}] &= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} t\, dt\right] = \mathrm{E}_u\left[\mathrm{E}\left[\sum_{k=1}^{\nu_i} \int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} t\, dt \mid \nu_i\right]\right] \\ &= \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \mathrm{E}\left[(\tau_{i,k}\, T_{i,k}^{S^c} + \frac{(T_{i,k}^{S^c})^2}{2})\middle|\nu_i\right]\right] \\ &= \mathrm{E}[T_i^{S^c}]\, \mathrm{E}_u\left[\sum_{k=1}^{\nu_i} \tau_{i,k}\right] + \frac{\mathrm{E}[\nu_i]\, \mathrm{E}[(T_i^{S^c})^2]}{2}. && (80) \end{aligned}$$

Note that equality (80) holds because, since policy u is nonanticipative, $\tau_{i,k}$ and $T_{i,k}^{S^c}$ are independent random variables. Hence, by (79) and (80):

$$\frac{z_i^u - \frac{1}{2}\mathrm{E}[\nu_i]\,\mathrm{E}[v_i^2]}{\mathrm{E}[v_i]} = \frac{\mathrm{E}_u[r_i^{\mathrm{II},S}] - \frac{1}{2}\mathrm{E}[\nu_i]\,\mathrm{E}[(T_i^{S^c})^2]}{\mathrm{E}[T_i^{S^c}]}, \quad i \in S \tag{81}$$

and thus we obtain:

$$\sum_{i\in S} A_i^S z_i^u = \mathrm{E}_u[\sum_{i\in S} r_i^{\mathrm{II},S}\Big] + \sum_{i\in S} b_i(S). \tag{82}$$

We will first show that generalized conservation law (26) holds. Consider a policy π that gives complete priority to S^c-jobs. Applying Proposition 4(a), we obtain:

$$\begin{aligned}\int_0^{T_m^E} t\,dt &= \int_0^{T_m^{S^c}} t\,dt + \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} t\,dt \\ &= \frac{(T_m^{S^c})^2}{2} + \sum_{i\in S} r_i^{\mathrm{II},S}. \end{aligned}\tag{83}$$

Hence, taking expectations and using equation (82) and the definition of $b(S)$ we obtain

$$\sum_{i\in S} A_i^S z_i^\pi = b(S),$$

which proves that generalized conservation law (26) holds.

We next show that generalized conservation law (27) is satisfied. Let the jobs be selected under admissible policy u. Then, Proposition 4 (part (b)) applies, and we can write

$$\int_0^{T_m^E} t\,dt = \int_0^{\Delta_m^S} t\,dt + \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+\Delta_{i,k}^S} t\,dt. \tag{84}$$

On the other hand, we have

$$\begin{aligned}\sum_{i\in S} r_i^{\mathrm{II},S} &= \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+T_{i,k}^{S^c}} t\,dt \\ &\geq \sum_{i\in S}\sum_{k=1}^{\nu_i}\int_{\tau_{i,k}}^{\tau_{i,k}+\Delta_{i,k}^S} t\,dt &(85)\\ &= \int_0^{T_m^E} t\,dt - \int_0^{\Delta_m^S} t\,dt &(86)\\ &\geq \int_0^{T_m^E} t\,dt - \int_0^{T_m^{S^c}} t\,dt. &(87)\end{aligned}$$

Notice that (85) follows by Proposition 4 (part (c)), (86) follows by (84), and (87) by Proposition 4 (part (c)). Hence, taking expectations in (87), and applying (82) we obtain

$$\sum_{i\in S} A_i^S z_i^u = \mathrm{E}_u[\sum_{i\in S} r_i^{\mathrm{II},S}] + \sum_{i\in S} b_i(S)$$

$$\begin{aligned} &\geq \mathrm{E}[\int_0^{T_m^E} t\,dt] - \mathrm{E}[\int_0^{T_m^{S^c}} t\,dt] + \sum_{i\in S} b_i(S) \\ &= b(S) \qquad (88) \end{aligned}$$

which proves that generalized conservation law (27) holds, and this completes the proof of the theorem. □

Corollary 3 *The performance space for branching bandits corresponding to the performance vector z^u is the extended polymatroid base $\mathcal{B}(A,b)$; furthermore, the vertices of $\mathcal{B}(A,b)$ are the performance vectors corresponding to the fixed priority rules.*

3.2.2 The Undiscounted Tax Problem

Let us associate with a branching bandit process the following linear tax structure: A holding tax C_i per unit time is incurred continuously during the stay of a type i job in the system. Let us denote

$V_u^{(0,C)}(m)$ = expected total tax incurred under policy u, given that there are initially m_i type i jobs in the system, for $i \in E$.

The tax problem is the following optimal control problem: find an admissible policy u^* that minimizes $V_u^{(0,C)}(m)$ over all admissible policies u. In this section we find a closed formula for $V_u^{(0,C)}(m)$ and show how to solve the problem using algorithm $\mathcal{A}_1$. For that purpose, we need some preliminary results:

Expected System Times

Let $Q_j^{*u}(\cdot)$, $I_j(\cdot)$ and $x_j^u(\cdot)$ be as in Subsection 3.1. By definition we have

$$Q_j^{*u}(0) = \int_0^\infty \mathrm{E}_u[Q_j(t)|Q(0) = m]\,dt, \quad j \in E$$

and

$$x_j^u(0) = \mathrm{E}_u\Big[\int_0^\infty I_j(t)\,dt|Q(0) = m\Big], \quad j \in E$$

From the above formulas, it is clear that

1. $Q_j^{*u}(0)$ is the expected total time spent in the system by type j jobs under policy u.
2. $x_j^u(0)$ is the expected total time spent working on type j jobs under policy u. Clearly, $x_j^u(0)$ does not depend on the policy u. Hence, we shall write $x_j(0) \equiv x_j^u(0)$.

Now, letting $\alpha \searrow 0$ on equation (60) we obtain

$$Q_j^{*u}(0) = -\frac{1}{\mathrm{E}[v_j]}\,(x_j^u)'(0) + \sum_{i\in E} \frac{\mathrm{E}[N_{ij}]}{\mathrm{E}[v_i]}(x_i^u)'(0) + h_j, \quad j \in E. \tag{89}$$

where

$$h_j = (1 - \frac{\mathrm{E}[v_j^2]}{2\mathrm{E}[v_j]^2})\, x_j(0) - \sum_{i\in E} \mathrm{E}[N_{ij}]\,(1 - \frac{\mathrm{E}[v_i^2]}{2\mathrm{E}[v_i]^2})\, x_i(0). \tag{90}$$

and $(x_j^u)'(0)$ denotes the right derivative of $x_j^u(\alpha)$ at $\alpha = 0$, that is:

$$(x_j^u(0))' = -\mathrm{E}_u\left[\int_0^\infty t\, I_j(t)\, dt\right] = -z_j^u.$$

Hence, we have

$$Q_j^{*u}(0) = \frac{1}{\mathrm{E}[v_j]}\, z_j^u - \sum_{i\in E} \frac{\mathrm{E}[N_{ij}]}{\mathrm{E}[v_i]} z_i^u + h_j, \quad j \in E. \tag{92}$$

Modelling and Solution of the Tax Problem

We have, by (92),

$$\begin{aligned} V_u^{(0,C)}(m) &= \sum_{i\in E} C_i Q_i^{*u}(0) \\ &= \sum_{i\in E} \left\{ \frac{C_i - \sum_{j\in E} \mathrm{E}[N_{ij}]\, C_j}{\mathrm{E}[v_i]} \right\} z_i^u + \sum_{i\in E} C_j h_j. \end{aligned} \tag{93}$$

From equation (93) it is straightforward to apply the results of Section 2 to solve the optimal control problem: use algorithm $\mathcal{A}_1$ with input $(\hat{R}, A)$, where

$$\hat{R}_i = \frac{C_i - \sum_{j\in E} \mathrm{E}[N_{ij}]\, C_j}{\mathrm{E}[v_i]}. \tag{94}$$

Let $\gamma_1, \ldots, \gamma_n$ be the corresponding generalized Gittins indices. Then we have the result

Theorem 11 (Optimality and Indexability: Undiscounted Branching Bandits) (a) *Algorithm $\mathcal{A}_1$ provides an optimal policy for the undiscounted tax branching bandit problem.*
(b) ***An optimal policy is to work at each decision epoch on a project with largest current index γ_i.***

3.2.3 Computation of A and $b(\cdot)$

In this section we compute the matrix A and the set function $b(\cdot)$ as follows. Recall that

$$A_i^S = \frac{\mathrm{E}[T_i^{S^c}]}{\mathrm{E}[v_i]}, \quad i \in S,$$

and

$$b(S) = \frac{1}{2}\mathrm{E}[\,(T_m^E)^2\,] - \frac{1}{2}\mathrm{E}[\,(T_m^{S^c})^2\,] + \sum_{i\in S} \frac{\mathrm{E}[\nu_i]\,\mathrm{E}[v_i^2]}{2}\left(\frac{\mathrm{E}[T_i^{S^c}]}{\mathrm{E}[v_i]} - \frac{\mathrm{E}[(T_i^{S^c})^2]}{\mathrm{E}[v_i^2]}\right).$$

From equation (65) we obtain, taking expectations:

$$\mathrm{E}[T_i^S] = \mathrm{E}[v_i] + \sum_{j\in S} \mathrm{E}[N_{ij}]\,\mathrm{E}[T_j^S], \quad i \in S. \tag{95}$$

Solving this linear system we obtain $\mathrm{E}[T_i^S]$. Note that the computation of A_i^S is much easier in the undiscounted case compared with the discounted case, where we had to solve a system of nonlinear equations. Also, applying the conditional variance formula to (65) we obtain:

$$\mathrm{Var}[T_i^S] = \mathrm{Var}[v_i] + (\mathrm{E}[T_j^S])_{j\in S}^T \,\mathrm{Cov}\left[(N_{ij})_{j\in S})\right]\,(\mathrm{E}[T_j^S])_{j\in S} + \sum_{j\in S} \mathrm{E}[N_{ij}]\,\mathrm{Var}[T_j^S], \, i \in S. \tag{96}$$

Solving this linear system we obtain $\mathrm{Var}[T_i^S]$ and thus $\mathrm{E}[(T_i^S)^2]$. Moreover,

$$\mathrm{E}[\nu_j] = m_j + \sum_{i\in E} \mathrm{E}[N_{ij}]\,\mathrm{E}[\nu_i], \quad j \in E. \tag{97}$$

Finally, from equation (69) we obtain

$$\mathrm{E}[T_m^S] = \sum_{i\in S} m_i\,\mathrm{E}[T_i^S], \tag{98}$$

and

$$\mathrm{Var}[T_m^S] = \sum_{i\in S} m_i\,\mathrm{Var}[T_i^S]. \tag{99}$$

3.2.4 Stability condition

We investigate in this section under what conditions, the linear systems (95) and (96) have a positive solution for all sets $S \subseteq E$. In this way we can address the stability of a branching bandits process, in the sense that the first two moments of a busy period of a branching bandit process are finite. Let N denote the matrix of $E[N_{ij}]$.

Theorem 12 (Stability of branching bandits) *The branching bandits process is stable if and only if the matrix $I - N$ is positive definite.*

Proof

Suppose $I - N$ is positive definite. We will show the system is stable. System (95) can be written in vector notation as follows:

$$(I - N)_S T_S = v_S, \tag{100}$$

where $T_S = (E[T_i^S])_{i\in S}$. Solving the system using Cramer's rule and expanding the determinant in the nominator along the column v_S we obtain:

$$E[T_i^S] = \frac{\sum_{r\in S, r\neq i} \theta_r v_r + v_i det[(I-N)_{S\setminus\{i\}}]}{det[(I-N)_S]}, \tag{101}$$

where θ_r are nonegative numbers (which are determinants themselves). If $I - N$ is positive definite, then $det[(I-N)_S] > 0$ for all $S \subseteq E$ and thus system (95) has a solution $E[T_i^S] > 0$ for all $i \in S$ and $S \subseteq E$. Similarly, (96) can be written as

$$(I-N)_S x_S = u_S,$$

where $x_S = (Var[T_i^S])_{i\in S}$ and $u_S \geq 0$. Therefore, using the same argument it follows that if $I - N$ is positive definite, then $Var[T_i^S] \geq 0$. Hence, from (98) and (99) we obtain that the first two moments of the busy periods are finite, i.e., the system is stable.

Conversely, if the system is stable, we will show that $I - N$ is positive definite. Since the system is stable for all initial vectors m, it follows that $E[T_i^S]$ have finite nonegative values for all $i \in S$ and $S \subseteq E$, i.e., system (100) has a positive solution for all $S \subseteq E$. We will show by induction on $|S|$ that $det[(I-N)_S] > 0$ for all $S \subseteq E$. For $|S| = 1$, $E[T_i^i] = \frac{v_i}{det[(I-N)_i]} > 0$, which implies that $det[(I-N)_i] > 0$. Assuming that the induction hypothesis is true for $|S| = k$, we use (101) to obtain:

$$det[(I-N)_S] = \frac{\sum_{r\in S, r\neq i} \theta_r v_r + v_i det[(I-N)_{S\setminus\{i\}}]}{E[T_i^S]} > 0,$$

from the induction hypothesis. Therefore, $I - N$ is positive definite. □

Note that the condition $N < I$ ($I-N$ positive definite) naturally generalizes the stability condition $\rho < 1$ in queueing systems as follows: If we interpret a queueing system as a branching bandit then $N < I$ translates to $E[N] = \rho = \lambda E[v] < 1$, since N is the number of customers that arrive (at a rate λ) during the service time v of a customer.

3.3 Relation between Discounted and Undiscounted Tax Problem

In this subsection we study the asymptotic behaviour of the optimal policies in the discounted tax problem as the discount factor α approaches 0, and its relation with the undiscounted tax problem, that corresponds to α equal to 0. It is easy to see that, using the notation of Subsections 3.1 and 3.2, that

$$\lim_{\alpha\searrow 0} A_{i,\alpha}^S = A_i^S, \tag{102}$$

and

$$\begin{aligned}\lim_{\alpha\searrow 0}\alpha\,\hat{R}_{i,\alpha} &= \lim_{\alpha\searrow 0}\Big\{C_i-\sum_{j\in E}\mathrm{E}[N_{ij}]\,C_j\Big\}\frac{\alpha\,\Psi(\alpha)}{1-\Psi(\alpha)}\\ &= \frac{C_i-\sum_{j\in E}\mathrm{E}[N_{ij}]\,C_j}{\mathrm{E}[v_i]}=\hat{R}_i. \qquad (103)\end{aligned}$$

Therefore, because of the structure of the generalized Gittins indices (see Proposition 3) it follows from (102) and (103) that the generalized Gittins indices of the undiscounted and discounted tax problem are related as follows:

$$\lim_{\alpha\searrow 0}\alpha\,\gamma_i(\alpha)=\gamma_i. \qquad (104)$$

A consequence of (104) is that a policy which is asymptotically optimal in the discounted tax problem for $\alpha \searrow 0$ will be optimal for the undiscounted problem.

4 Applications

In this section we apply the previous theory to several classical stochastic scheduling problems.

4.1 The Multi-armed Bandit Problem

The multi-armed bandit problem was defined in the introduction.

There are K parallel projects, indexed k = 1, ..., K. Project k can be in one of a finite number of states $i_k \in E_k$. At each instant of discrete time $t = 0, 1, \ldots$ one can work on only a single project. If one works on project k in state $i_k(t)$ at time t then one receives an immediate expected reward of $R_{i_k(t)}$. Rewards are additive and discounted in time by a factor β. The state $i_k(t)$ changes to $i_k(t+1)$ by a Markov transition rule (which may depend on k, but not on t), while the states of the projects one has not engaged remain unchanged, i.e., $i_l(t+1) = i_l(t)$ for $l \neq k$. Let $P^k = (p^k_{ij})_{i,j\in E_k}$ be the matrix of Markov transition probabilities corresponding to project k. The problem is how to allocate one's resources to projects sequentially in time in order to maximize expected total discounted reward over an infinite horizon. That is, if $j(t)$ denotes the state of the project engaged at time t, the goal is to find a nonidling and nonanticipative scheduling policy u that maximizes

$$\mathrm{E}_u[\sum_{t=0}^{\infty}\beta^t R_{j(t)}]. \qquad (105)$$

We model the problem as a branching bandits problem in order to apply the results of the previous section. For this reason we set $e^{-\alpha} = \beta$, $v_i \equiv 1$. We also define matrix $P = (p_{ij})_{i,j \in E}$ by

$$p_{ij} = \begin{cases} p_{ij}^k, & \text{if } i, j \in E_k, \text{ for some } k = 1, \ldots, K; \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, by (62) we obtain:

$$\begin{aligned} \Phi_i(\alpha, z_1, \ldots, z_n) &= \mathrm{E}[e^{-\alpha v_i} z_1^{N_{i1}} \ldots z_n^{N_{in}}] \\ &= e^{-\alpha} \sum_{j \in E} p_{ij} z_j \\ &= \beta \sum_{j \in E_k} p_{ij} z_j, \qquad \text{for } i \in E_k \end{aligned} \tag{106}$$

and, by (66)

$$\begin{aligned} \Psi_i^S(\alpha) &= \Phi_i\Big(\alpha, (\Psi_j^S(\alpha))_{j \in S}, 1_{S^c}\Big) \\ &= \beta \Big\{ \sum_{j \in S} p_{ij} \Psi_j^S(\alpha) + \sum_{j \in S^c} p_{ij} \Big\} \\ &= \beta \Big\{ 1 - \sum_{j \in S} p_{ij} (1 - \Psi_j^S(\alpha)) \Big\}, \qquad \text{for } i \in E. \end{aligned} \tag{107}$$

By introducing

$$t_i^S = \frac{1 - \Psi_i^S(\alpha)}{1 - \Psi_i(\alpha)}, \qquad \text{for } i \in S,$$

and noticing that since $v_i = 1$, $\Psi_i(\alpha) = \beta$, it follows from (107) that

$$t_i^S = 1 + \beta \sum_{j \in S} p_{ij} t_j^S, \qquad i \in S, \tag{108}$$

and by (107) and Proposition 6 we obtain

$$A_{i,\alpha}^S = 1 + \beta \sum_{j \in S^c} p_{ij} t_j^{S^c}. \qquad i \in S. \tag{109}$$

Moreover, since $\Psi_j^E(\alpha) = 0$,

$$\begin{aligned} b_\alpha(S) &= \frac{1}{\alpha} \prod_{j \in S^c} [\Psi_j^{S^c}(\alpha)]^{m_j} \\ &= \frac{1}{\alpha} \prod_{j \in S^c} (t_j^{S^c})^{m_j} \end{aligned} \tag{110}$$

where

$$m_j = \begin{cases} 1, & \text{if at time } t = 0 \text{ there is a bandit in state } j; \\ 0, & \text{otherwise.} \end{cases}$$

The structure of the matrix $P = (p_{ij})$ implies that

$$A^S_{j,\alpha} = A^{S \cap E_k}_{j,\alpha}, \qquad \text{for } j \in S \cap E_k$$

which implies that the index decomposition condition (17) holds, and therefore Theorem 3 applies, giving a new proof of Gittins theorem:

Theorem 13 (Gittins and Jones [14]) *For each project k there exist indices $\{\gamma_i^k\}_{i \in E_k}$, depending only on characteristics of project k, such that an optimal policy is to engage at each time a project with largest current index.*

By the results of Subsection 3.1.3 we know that the generalized Gittins indices for this bandit problem coincide with the usual Gittins indices. Further, by definition of generalized Gittins indices, we obtain a characterization of Gittins indices as sums of dual variables, a purely algebraic characterization. Also, note that Theorem 13 implies that the multi-armed bandit problem not only has an optimal index policy, but it has an optimal index policy which satisfies the stronger index decomposition property, as described in Subsection 2.4. By Theorem 6, the Gittins indices can be computed by solving K subproblems, applying algorithm $\mathcal{A}_1$ to subproblem k, with $|E_k|$ job types, $k = 1, \ldots, K$. It is easy to verify the following complexity result:

Proposition 7 *The complexity of algorithm $\mathcal{A}_1$ applied to subproblem k for computing the Gittins indices of project k is*

$$O(|E_k|^3).$$

The algorithm proposed by Varaiya, Walrand and Buyukkoc [33] has the same time complexity as algorithm $\mathcal{A}_1$. In fact, both algorithms are closely related, as we will see next Let t_i^S be as given by (108). Let r_i^S be given by

$$r_i^S = R_i + \beta \sum_{j \in S} p_{ij} r_j^S, \quad i \in S.$$

Let us now state the algorithm of Varaiya, Walrand and Buyukkoc:

Algorithm VWB:

Step 0. Pick $\pi_n \in \operatorname{argmax} \{ \frac{r_i^{\{i\}}}{t_i^{\{i\}}} : i \in E \}$; let $g_{\pi_n} = \max\{ \frac{r_i^{\{i\}}}{t_i^{\{i\}}} : i \in E \}$; set $J_n = \{\pi_n\}$.

Step k. For $k = 1, \ldots, n-1$:

pick $\pi_{n-k} \in \operatorname{argmax} \{ \frac{r_i^{J_{n-k} \cup \{i\}}}{t_i^{J_{n-k} \cup \{i\}}} : i \in E \setminus J_{n-k} \}$; set $g_{\pi_{n-k}} = \{ \frac{r_i^{J_{n-k} \cup \{i\}}}{t_i^{J_{n-k} \cup \{i\}}} : i \in E \setminus J_{n-k} \}$;
set $J_{n-k} = J_{n-k+1} \cup \{\pi_{n-k}\}$.

Varaiya *et al.* [33] proved that $g_1, \ldots, g_n$, as given by algorithm $\mathcal{VWB}$, are the Gittins indices of the multi-armed bandit problem. Let $(\pi, \overline{y}, \nu, \mathcal{S})$ be an output of algorithm $\mathcal{A}_1$. We state, without proof, the following relation between algorithms $\mathcal{A}_1$ and $\mathcal{VWB}$:

Proposition 8 *The following relations hold: For* $j = 2, \ldots, n$

$$\frac{r_i^{\{\pi_j, \ldots, \pi_n\} \cup \{i\}}}{t_i^{\{\pi_j, \ldots, \pi_n\} \cup \{i\}}} - \frac{R_i - \sum_{l=j}^{n} A_i^{\{\pi_1, \ldots, \pi_l\}} \nu_l}{A_i^{\{\pi_1, \ldots, \pi_{j-1}\}}} \equiv \frac{r_{\pi_j}^{\{\pi_j, \ldots, \pi_n\}}}{t_{\pi_j}^{\{\pi_j, \ldots, \pi_n\}}}, i \in \{\pi_1, \ldots, \pi_{j-1}\} \tag{111}$$

and

$$\frac{r_i^{\{i\}}}{t_i^{\{i\}}} - \frac{R_i}{A_i^E} \equiv 0, \quad i \in E, \tag{112}$$

and therefore, algorithms $\mathcal{A}_1$ *and* $\mathcal{VWB}$ *are equivalent.*

4.2 Scheduling Control of a Multiclass Queue with Bernoulli Feedback

Klimov [22] introduced the following queueing scheduling process: There is a single server and n customer types. External arrivals of type i customers form a Poisson process of rate λ_i, for $i \in E = \{1, \ldots, n\}$. Service times for type i customers are independent and identically distributed as a random variable v_i with distribution function $G_i(\cdot)$. When service of a type i customer is completed, the customer either joins the queue of type j customers, with probability p_{ij} (thus becoming a type j customer), or with probability $1 - \sum_{j \in E} p_{ij}$ leaves the system. The server selects the jobs according to an admissible policy u; the decision epochs are $t = 0$ (if there is initially some customer present), the epochs when a customer arrives to find the system empty and the epochs when a customer completes service (and some customer remains in the system). Let us consider the following three classes of admissible policies: $\mathcal{U}$ is the class of all nonidling, nonpreemptive and nonanticipative policies; $\mathcal{U}_0$ is the class of all nonpreemptive and nonanticipative policies (idleness is allowed); and $\mathcal{U}^p$ is the class of all nonidling and nonanticipative policies (preemption is allowed).

Klimov [22] solved, by direct methods, the associated optimal control problem over $\mathcal{U}$ with a time-average holding cost criterion. Harrison [18] solved, using dynamic programming, the optimal control problem over $\mathcal{U}_0$ with a discounted reward-cost criterion, in the special case that there is no feedback. Tcha and Pliska [29] extended Harrison's results to the case with feedback. They also solved the control problem over $\mathcal{U}^p$, in the case that the service times are exponential.

The Discounted Case

Let us consider the following reward-cost structure: There is a continuous holding cost C_i per unit time for each type i customer staying in the system, and an instantaneous reward of R_i at the epoch of completion of service of a type i customer. There is also an instantaneous reward of idleness R_0 at the end of an idle period. All costs and rewards are discounted in time by a discount factor $\alpha > 0$. The optimal control problem is to find an admissible policy to schedule the server so as to maximize the expected total discounted reward minus holding cost over an infinite horizon. Let us denote $P_{\mathcal{U}}$. $P_{\mathcal{U}_0}$ and $P_{\mathcal{U}^p}$ the optimal control problems corresponding to the classes of admissible policies $\mathcal{U}$. $\mathcal{U}_0$ and $\mathcal{U}^p$, respectively. We will model each of these problems as a branching bandit problem. We will also prove, applying the Index Decomposition Theorem, that in order to solve problem $P_{\mathcal{U}_0}$ we only need to solve problem $P_{\mathcal{U}}$.

First, let us consider problem $P_{\mathcal{U}}$. This problem can be modelled as a branching bandit problem with n job types, as follows: We interpret the customers as jobs. The descendants N_{ij} of a type i job are composed of the transition of the job to another type (or outside the system) and of the external Poisson arrivals. The transform $\Phi_i(.)$ is given by

$$\begin{aligned}
\Phi_i(\alpha, z_1, \ldots, z_n) &= \mathrm{E}[e^{-\alpha v_i} z_1^{N_{i1}} \ldots z_n^{N_{in}}] \\
&= \mathrm{E}\left[(1 - \sum_{j\in E} p_{ij}(1 - z_j))e^{-v_i(\alpha + \sum_{j\in E} \lambda_j(1-z_j))}\right] \\
&= \left\{(1 - \sum_{j\in E} p_{ij}(1 - z_j))\Psi_i(\alpha + \sum_{j\in E} \lambda_j(1 - z_j)\right\}, \quad i \in E. \qquad (113)
\end{aligned}$$

Also, by (66) and (113)

$$\Psi_i^S(\alpha) = \left\{1 - \sum_{j\in S} p_{ij}(1 - \Psi_j^S(\alpha)\right\} \Psi_i\left[\alpha + \sum_{j\in S} \lambda_j(1 - \Psi_j^S(\alpha))\right]. \quad i \in E. \qquad (114)$$

Let $x^u(\alpha) = (x_1^u(\alpha), \ldots, x_n^u(\alpha))^T$ denote the performance vector. as in Section 3.1. We know that $x^u(\alpha)$ satisfies generalized conservation laws. By Proposition 6. the corresponding matrix A_α is given by

$$A_{i,\alpha}^S = \frac{1 - \Psi_i^{S^c}(\alpha)}{1 - \Psi_i(\alpha)}, \quad i \in S.$$

Let us consider now problem $P_{\mathcal{U}_0}$. In order to model the option of idleness. we modify the previous branching bandit process by adding an *idling job type* , which we denote 0. The duration of job type 0, v_0, is exponentially distributed with parameter $\lambda = \lambda_1 + \cdots + \lambda_n$ (since it models time until the next arrival); the N_{ij}, with $i, j \in E$, are as in the previous case. $N_{00} \equiv 0$; $N_{0i} \equiv 0$ and $N_{i0} \equiv 0$ for $i \in E$. It is easy to see that the corresponding transform $\overline{\Phi}_i(\cdot)$ satisfies

$$\overline{\Phi}_i(\alpha, z_0, z_1, \ldots, z_n) = \Phi_i(\alpha, z_1, \ldots, z_n), \quad i \in E.$$

Hence, it follows that

$$\overline{\Psi}_i^{S\cup\{0\}}(\alpha) = \Psi_i^S(\alpha), \quad i \in E, \quad S \subseteq E,$$

and

$$\overline{\Psi}_0^{S\cup\{0\}}(\alpha) = \overline{\Psi}_0^{\{0\}}(\alpha), \quad S \subseteq E.$$

Consequently, we have, for $i \in S \subseteq E$ that

$$\overline{A}_{i,\alpha}^{S} = A_{i,\alpha}^{S},$$

$$\overline{A}_{i,\alpha}^{S\cup\{0\}} = A_{i,\alpha}^{S},$$

and

$$\overline{A}_{0,\alpha}^{S\cup\{0\}} = 1 = \overline{A}_{0,\alpha}^{\{0\}}.$$

Therefore, condition (17) holds, and the Index Decomposition Theorem 6 applies. Now, we have

$$\mathrm{E}[N_{ij}] = p_{ij} + \lambda_j \mathrm{E}[v_i],$$

and

$$\overline{\Psi}_0(\alpha) = \frac{\lambda}{\lambda + \alpha}.$$

By (56)

$$\overline{V}_u^{(R,C)}(m) = \sum_{i\in E}\Big\{R_i + \frac{C_i - \sum_{j\in E}(p_{ij} + \lambda_j \mathrm{E}[v_i])C_j}{\alpha}\Big\}\frac{\alpha\Psi_i(\alpha)}{1 - \Psi_i(\alpha)}x_i^u + \lambda R_0 x_0^u, \quad u \in \mathcal{U}_0.$$

Hence the index of the subsystem composed of job type 0 is $\gamma_0 = \lambda R_0$. The indices γ_i, for $i \in E$, are computed from algorithm $\mathcal{A}_1$ applied to problem $P_{\mathcal{U}}$. Therefore, if $\gamma_1 \leq \cdots \leq \gamma_{i^*-1} \leq \gamma_0 \leq \gamma_{i^*} \leq \cdots \gamma_n$ then an optimal policy is to serve customers of types $i^*, \ldots, n$ with a fixed priority policy, giving highest priority to n, and never serve customer types $1, \ldots, i^* - 1$. That is, the optimal policy is a *modified static policy*, as proved by Harrison [18] and Tcha and Pliska [29].

The Preemptive Case

In preemption is allowed, then the decision epochs are the arrival epochs as well as the departures epochs of customers. If the service time v_i is exponential with rate μ_i, for $i \in E$, then it is easy to model the possibility of ***preemption***: model the process as a branching bandit process with n job types. Job type i has a duration $\hat{v}_i$ exponentially distributed with rate $\hat{\mu}_i = \mu_i + \lambda$, where $\lambda = \lambda_1 + \cdots + \lambda_n$. As for the descendants of a type i job, there are three cases: (1) One descendant, of type j with probability $\frac{\mu_i}{\hat{\mu}_i} p_{ij}$ (corresponding to the case that service of the type i customer ends before any arrival occurs and the customer moves to queue j); (2) two descendants, one of type i and the other of type j with probability $\frac{\lambda_j}{\hat{\mu}_i}$ (corresponding to the case that a type j customer arrives before service of the type i customer is completed); and (3) no descendants, with probability $\frac{\mu_i}{\hat{\mu}_i}(1 - \sum_{j \in E} p_{ij})$ (corresponding to the case that service of the type i customer ends before any arrival occurs, and the customer moves out of the system).

The Undiscounted Case: Klimov's Problem

Klimov [22] first considered the problem of optimal control of a single-server multiclass queue with Bernoulli feedback, with the criterion of minimizing the time average holding cost per unit time. He proved that the optimal nonidling, nonpreemptive and nonanticipative policy is a fixed priority policy, and presented an algorithm for computing the priorities (starting with the lowest priority type and ending with the highest priority). Tsoucas [32] modelled Klimov's problem as an optimization problem over an extended polymatroid using as performance measures

$$L_i^u = \text{time average length of queue } i \text{ under policy } u.$$

Algorithm $\boldsymbol{\mathcal{A}_1}$ applied to this problem is exactly Klimov's original algorithm. A disadvantage in this case is that priorities are computed from lowest priority to highest priority. Also, Tsoucas does not obtain closed form formulae for the right hand sides of the extended polymatroid, so it is not possible to evaluate the performance of an optimal policy. Our approach gives explicit formulae for all of the parameters of the extended polymatroid and also explains the somewhat surprising property that the optimal priority rule does not depend in this case on the arrival rates. The key observation is that an optimal policy under the time average holding cost criterion also minimizes the expected total holding cost in each busy period (see Nain *et al.* [24] for further discussion). Now, we may model the first busy period of Klimov's problem as a branching bandit process with the undiscounted tax criterion, as considered in Section 3.

Assuming that the system is stable, we apply the results of Subsection 3.2. We define $\mu_i = \mathrm{E}[v_i]$ and $t_i^S = \mathrm{E}[T_i^S]$. By (65) we have

$$t_i^S = \mu_i + \sum_{j \in S} (p_{ij} + \mu_i \lambda_j) t_j^S, \quad i \in E, \tag{115}$$

which in vector notation becomes:

$$t_{S^c}^{S^c} = \mu_{S^c} + (P_{S^c,S^c} + \mu_{S^c} \lambda_{S^c}^T)\, t_{S^c}^{S^c},$$

i.e.,

$$t_{S^c}^{S^c} = (I_{S^c} - P_{S^c,S^c} - \mu_{S^c} \lambda_{S^c}^T)^{-1}\, \mu_{S^c}.$$

and

$$t_S^{S^c} = \mu_S + (P_{S,S^c} + \mu_S \lambda_{S^c}^T) t_{S^c}^{S^c}.$$

After algebraic manipulations we obtain

$$t_i^{S^c} = \left(\mu_i + p_{i,S^c}^T\, (I_{S^c} - P_{S^c,S^c})^{-1}\, \mu_{S^c}\right) \frac{\det(I_{S^c} - P_{S^c,S^c})}{\det(I_{S^c} - P_{S^c,S^c} - \mu_{S^c} \lambda_{S^c}^T)}. \quad i \in S. \tag{116}$$

Therefore, by definition of A_i^S in (73) we find that $A_i^S = t_i^{S^c}/\mu_i$, for $i \in S$. while $b(S)$ is given by (74). Now, letting

$$K_S = \frac{\det(I_{S^c} - P_{S^c,S^c})}{\det(I_{S^c} - P_{S^c,S^c} - \mu_{S^c} \lambda_{S^c}^T)},$$

we may define $\hat{A}_i^S = A_i^S / K_S$, and $\hat{b}(S) = b(S)/K_S$, thus eliminating the dependence on the arrival rates of matrix $\hat{A}$. As for the objective function, we have by (93):

$$V_{\boldsymbol{u}}^{(0,C)} = \sum_{i \in E} \left\{ \frac{C_i - \sum_{j \in E} p_{ij}\, C_j}{\mu_i} \right\} z_i^u - b(E) \sum_{j \in E} C_j \lambda_j + \sum_{i \in E} C_j h_j. \tag{117}$$

Hence the problem can be solved by applying algorithm $\mathcal{A}_1$ with input (R, A), where

$$R_i = \frac{C_i - \sum_{j \in E} p_{ij} C_j}{\mu_i}, \quad i \in E,$$

and since $(R, \hat{A})$ do not depend on the arrival rates neither does the optimal policy. Note that as opposed to Klimov's algorithm, with this algorithm priorities are computed from highest to lowest. This top-down algorithm was first proposed by Nain *et al.* [24], who proved its optimality using interchange arguments. Bhattacharya *et al.* [3] provided a direct optimality proof. Nain *et al.* proved that the resulting optimal index rule is also optimal among idling policies for general service time distributions, and among preemptive policies when the service time distributions are exponential. It is also easy to verify these facts using our approach (in particular, the index of the idling state is 0, whereas all other indices are nonnegative).

Moreover, in the case that the arriving jobs are divided into K projects, where a type k project consists of jobs with types in a finite set E_k, jobs in E_k can only make transitions within E_k, and E is partitioned as $E = \cup_{k=1}^{K} E_k$, then it is easy to see that the Index Decomposition Theorem 6 applies, and therefore we can decompose the problem into K smaller subproblems.

4.3 Multiclass Queueing Systems

Shantikumar and Yao [26] showed that a large variety of multiclass queueing systems satisfy ***strong conservation laws.*** The reader is referred to their paper for a list of particular systems and performance measures that satisfy strong conservation laws. All their results correspond to the special case that the performance space $\mathcal{B}(A, b)$ is a polymatroid.

4.4 Job Scheduling Problems without Arrivals; Deterministic Scheduling

There are n jobs to be completed by a single server. Job i has a service requirement distributed as the random variable v_i, with moment generating function $\Psi_i(\cdot)$. It is immediate to model this job scheduling process as a branching bandit process in which jobs have no descendants. Let us consider first the discounted case: For $\alpha > 0$ it is clear by definition of $A_{i,\alpha}^{S}$, in (36), that $A_{i,\alpha}^{S} \equiv 1$, for $i \in S$. Therefore the performance space of the vectors $x^u(\alpha)$ studied in Section 3 is a polymatroid. Consider the discounted reward-tax problem discussed in Section 3, in which a instantaneous reward R_i is received at the completion of job i, and a holding tax C_i is incurred for each unit of time that job i is in the system. Rewards and taxes are discounted in time with discount factor α. By (56) it follows that the generalized Gittins index for job i, in the problem of maximizing rewards minus taxes, is

$$\gamma_i(\alpha) = \left\{ R_i + \frac{C_i}{\alpha} \right\} \frac{\alpha \Psi_i(\alpha)}{1 - \Psi_i(\alpha)}. \tag{118}$$

Let us consider now the undiscounted case in the case without rewards. By definition of A_i^S in (73) we have $A_i^S \equiv 1$, for $i \in S$. Hence the performance space of the performance vectors z^u studied in Section 3 is also a polymatroid. Thus by equation (93) it follows that the generalized Gittins index for job i in the undiscounted tax problem is

$$\gamma_i = \frac{C_i}{\mathrm{E}[v_i]}, \tag{119}$$

thus providing a new polyhedral proof of the optimality of Smith's rule (see Smith [27]).

In the case that there are precedence constraints among the jobs that form out-trees. that is each job can have at most one predecessor, it is easy to see that the problem can also be modeled as a branching bandits problem and thus solvable using the theory we have developed in Section 3.

5 Reflections

We presented a unified treatment of several classical problems in stochastic and dynamic scheduling using polyhedral methods that leads, we believe, to a deeper understanding of their structural and algorithmic properties. Perhaps the most important idea we used is to ask the question: What is the performance space of a stochastic scheduling problem? We believe that the approach of characterizing the feasible region of a stochastic scheduling problem will lead to important new insights and methods and will bridge the artificial gap between applied probability and mathematical optimization. Indeed. we hope that our results will be of interest to applied probabilists, as they provide new interpretations. proofs, algorithms, insights and connections to important problems in stochastic scheduling, as well as to discrete optimizers, since they reveal a new fundamental structure (extended polymatroids) which has a genuinely applied origin.

References

[1] C. Bell, (1971), "Characterization and computation of optimal policies for operating an $M/G/1$ queue with removable server", *Operations Research*, **19**, 208-218.

[2] D. Bertsimas, I. Paschalidis and J. Tsitsiklis, (1992), "Optimization of multiclass queueing networks: polyhedral and nonlinear characterizations of achievable performance", working paper, Operations Research Center, MIT.

[3] P. P. Bhattacharya, L. Georgiadis and P. Tsoucas, (1991), "Problems of adaptive optimization in multiclass $M/GI/1$ queues with Bernoulli feedback". Paper presented in part at the ORSA/TIMS *Conference on Applied Probability in the Engineering. Information and Natural Sciences*, January 9-11, 1991, Monterey, California.

[4] P. P. Bhattacharya, L. Georgiadis and P. Tsoucas, (1991), "Extended polymatroids: Properties and optimization", *Proceedings of International Conference on Integer Programming and Combinatorial Optimization (Carnegie Mellon University)*. Mathematical Programming Society, 298-315.

[5] Y. R. Chen and M. N. Katehakis, (1986), "Linear programming for finite state multi-armed bandit problems", *Mathematics of Operations Research*, **11**, 183.

[6] A. Cobham, (1954), "Priority assignment in waiting line problems". *Operations Research*, **2**, 70-76.

[7] E. Coffman and I. Mitrani, (1980), "A characterization of waiting time performance realizable by single server queues", *Operations Research*, **28**, 810-821.

[8] E. Coffman, M. Hofri and G. Weiss, (1989), "Scheduling stochastic jobs with a two point distribution on two parallel machines", *Probab. Engin. Infor.*. to appear.

[9] D. R. Cox and W. L. Smith, (1961), *Queues*, Methuen (London) and Wiley (New York).

[10] J. Edmonds, (1970), "Submodular functions, matroids and certain polyhedra", in *Combinatorial Structures and Their Aplications*, 69-87. R. Guy *et al.* (eds.). Gordon & Breach, New York.

[11] A. Federgruen and H. Groenevelt, (1988), "Characterization and optimization of achievable performance in general queueing systems", *Operations Research*, **36**, 733-741.

[12] A. Federgruen and H. Groenevelt, (1988), "$M/G/c$ queueing systems with multiple customer classes: Characterization and control of achievable performance under non-preemptive priority rules", *Management Science*, **34**, 1121-1138.

[13] E. Gelenbe and I. Mitrani, (1980), *Analysis and Synthesis of Computer Systems*. Academic Press, New York.

[14] J. C. Gittins and D. M. Jones, (1974), "A dynamic allocation index for the sequential design of experiments". In J. Gani, K. Sarkadi & I. Vince (eds.), *Progress in Statistics European Meeting of Statisticians 1972*, vol. 1. Amsterdam: North-Holland, 241-266.

[15] J. C. Gittins, (1979), "Bandit processes and dynamic allocation indices". *Journal of the Royal Statistical Society Series*, **B 14**, 148-177.

[16] J. C. Gittins, (1989), *Bandit Processes and Dynamic Allocation Indices*. John Wiley.

[17] K. D. Glazebrook, (1987), "Sensitivity analysis for stochastic scheduling problems". *Mathematics of Operations Research*, **12**, 205-223.

[18] J. M. Harrison, (1975), "A priority queue with discounted linear costs". *Operations Research*, **23**, 260-269.

[19] J. M. Harrison, (1975), "Dynamic scheduling of a multiclass queue: discount optimality", *Operations Research*, **23**, 270-282.

[20] M. N. Katehakis and A. F. Veinott, (1987). "The multiarmed bandit problem: decomposition and computation", *Mathematics of Operations Research*, **12**, 262-268.

[21] L. Kleinrock, (1976), *Queueing Systems*. vol. 2. John Wiley.

[22] G. P. Klimov, (1974), "Time sharing service systems I", *Theory of Probability and Applications*, **19**, 532-551.

[23] E. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan and D.B. Shmoys, (1989), "Sequencing and scheduling; algorithms and complexity". Report BS-R8909, Centre for Mathematics and Computer Science, Amsterdam.

[24] P. Nain, P. Tsoucas and J. Walrand, (1989), "Interchange arguments in stochastic scheduling", *Journal of Applied Probability*, **27**, 815-826.

[25] K. W. Ross and D. D. Yao (1989), "Optimal dynamic scheduling in Jackson Networks", *IEEE Transactions on Automatic Control*, **34**, 47-53.

[26] J. G. Shanthikumar and D. D. Yao, (1992), "Multiclass queueing systems: Polymatroidal structure and optimal scheduling control", *Operations Research*, **40**. Supplement 2, S293-299.

[27] W. E. Smith, (1956), "Various optimizers for single-stage production", *Naval Research Logistics Quarterly*, **3**, 59-66.

[28] S. Stidham, (1972), "$L = \lambda W$: A discounted analog and a new proof". *Operations Research*, **20**, 1115-1126.

[29] D.-W. Tcha and S. R. Pliska, (1977), "Optimal control of single-server queueing networks and multi-class $M/G/1$ queues with feedback", *Operations Research*, **25**. 248-258.

[30] J. N. Tsitsiklis, (1986), "A lemma on the multi-armed bandit problem". *IEEE Transactions on Automatic Control*, **31**, 576-577.

[31] J. N. Tsitsiklis, (1993), "A short proof of the Gittins index theorem", in preparation.

[32] P. Tsoucas, (1991), "The region of achievable performance in a model of Klimov", Technical Report RC16543, IBM T. J. Watson Research Center.

[33] P. P. Varaiya, J. C. Walrand and C. Buyukkoc, (1985), "Extensions of the multiarmed bandit problem: The discounted case", *IEEE Transactions on Automatic Control*, **30**. 426-439.

[34] R. Weber, (1992), "On the Gittins index for multiarmed bandits". *The Annals of Applied Probability*, **2**, 1024-1033.

[35] G. Weiss, (1988), "Branching bandit processes", *Probability in the Engineering and Information Sciences*, **2**, 269-278.

[36] P. Whittle (1980), "Multi-armed bandits and the Gittins index", *Journal of the Royal Statistical Society*, **42**, 143-149.

[37] P. Whittle, (1981), "Arm acquiring bandits". *Annals of Probability*. **9**, 284-292.

[38] P. Whittle, (1982), *Optimization Over Time*, vol. 1. John Wiley, Chichester. U.K.