Approximate Dynamic Programming by Linear Programming for Stochastic Scheduling

Mohamed Mostagir Nelson Uhan

1 Introduction

In stochastic scheduling, we want to allocate a limited amount of resources to a set of jobs that need to be serviced. Unlike in deterministic scheduling, however, the parameters of the system may be stochastic. For example, the time it takes to process a job may be subject to random fluctuations. Stochastic scheduling problems occur in a variety of practical situations, such as manufacturing, construction, and compiler optimization.

As in deterministic scheduling, the set of stochastic scheduling problems is incredibly large and diverse. One important class of models involves scheduling a fixed number of jobs on a fixed number of identical parallel machines while minimizing a given performance measure. The processing times of the jobs are assumed to follow some joint probability distribution. In addition, there may be precedence constraints, or interdependencies between the jobs that require certain jobs not be scheduled until others are completed. The deterministic counterpart to this class of problems has been studied extensively [KSW98]. A naïve approach to these problems would be to take the expected processing times and use the algorithms for the deterministic problems. Unfortunately, it is easy to construct examples that shows that this approach can produce solutions that are arbitrarily bad. Fortunately, however, this class of stochastic scheduling problems can be easily cast as a Markov decision process (MDP) and therefore can be attacked by dynamic programming methods.

Results for this class of stochastic scheduling problems are somewhat scattered, and have been obtained using a variety of methods. Rothkopf [R66] showed that for one machine without precedence constraints, an index rule minimizes the expected sum of weighted completion times for arbitrary processing time probability distributions. Möhring, Radermacher and Weiss [MRW84, MRW85] study the analytic properties of various classes of scheduling policies, and determine optimal policies for special cases. Möhring, Schulz, and Uetz [MSU99] developed approximation algorithms for a variety of stochastic

scheduling problems using techniques from combinatorial optimization. Bertsekas and Castañon [BC99] focus on a different class of stochastic scheduling problems, the quiz problem and its variations. They show how rollout algorithms can be implemented efficiently, and present experimental evidence that the performance of rollout policies is near-optimal.

For this project, we consider a problem with one machine and an arbitrary normalized regular and additive objective function. We recast our finite-horizon decision problem into a stochastic shortest path problem. We show that for a relaxed formulation of our problem, the error of the solution obtained by the approximate linear programming approach to dynamic programming as presented in de Farias and Van Roy [dV03] is uniformly bounded over the number of jobs that need to be scheduled, provided that the expected job processing times are finite. Finally, we argue using results from dynamic programming that the approximate solution for the relaxed formulation of our problem is also not that far away from the optimal solution of the original problem.

2 The problem

Consider the following problem. We have a set of jobs $N = \{1, \ldots, n\}$ to be processed on one machine. The machine can only process one job at a time. The processing time of job i follows a discrete probability distribution p_i, and the distributions $p_1, \ldots, p_n$ are assumed to be pairwise stochastically independent. The jobs have to be scheduled nonpreemptively, that is, once a job has started, it must be processed continuously until it is finished. The schedule must also respect any precedence constraints, or interdependencies between jobs that require certain jobs not be scheduled before others are completed. We would like to minimize the expected value of the objective function

$$\gamma\left(C^{(1)}, \ldots, C^{(n)}\right) = \frac{1}{n} \sum_{i=0}^{n-1} h\left(R_i\right)\left(C^{(i+1)} - C^{(i)}\right)$$

where $C^{(i)}$ denotes the time of the ith job completion ($C^{(0)} = 0$), R_i is the set of jobs remaining to be processed at the time of the ith job completion, and h is a set function such that $h\left(\emptyset\right) = 0$. Such an objective function is said to be *additive.* The function h can be interpreted as the holding cost per unit time on the set of uncompleted jobs. We also assume that γ is nondecreasing in the job completion times.

Many common objective functions in scheduling are nondecreasing and additive. For example, $h\left(S\right) = \sum_{j \in S} w_j$ for all $S \subset N$ generates the normalized sum of weighted completion times, $\frac{1}{n} \sum_{j \in N} w_j C_j$.

2.1 Formulation as a finite horizon MDP

We can formulate this problem as a finite horizon MDP with a finite state space $\mathcal{S}$. For each state $x \in \mathcal{S}$, there is a set of available actions $\mathcal{A}_x$. Taking action

$a \in \mathcal{A}_x$ in state x, we transition to state y with probability $P_a(x,y)$ and incur cost $g_a(x,y)$.

Since we only have one machine, the state of the system x is sufficiently characterized by the set of jobs remaining to be processed R_x and the maximum completion time of the jobs completed so far, $C_{max}(x)$:

$$x = (C_{max}(x), R_x) \in \mathcal{S}.$$

The size of the state space is exponential in the number of jobs.

Since the objective function is nondecreasing in completion times, and since the jobs' processing times are independent of their start times, any optimal policy need not leave the machine deliberately idle at any time. Therefore, an action in our problem is simply the next job to be processed: at every state x, the action set is $\mathcal{A}_x \subset R_x$. If there are no precedence constraints, $\mathcal{A}_x = R_x$. The time stage costs are

$$g_a(x,y) = \frac{1}{n} h(R_x) (C_{max}(y) - C_{max}(x)).$$

The transition probabilities are

$$P_a(x,y) = \begin{cases} p_a(t) & \text{if } R_y = R_x \setminus \{a\} \text{ and } C_{max}(y) = C_{max}(x) + t \\ 0 & \text{otherwise.} \end{cases}$$

The problem is then to solve for the following finite-horizon cost-to-go function

$$J^*(x,n) = 0$$

$$J^*(x,t) = \min_{a \in \mathcal{A}_x} \left\{ \sum_{y \in \mathcal{S}} P_a(x,y) (g_a(x,y) + J^*(y,t+1)) \right\}, \quad t = 0,1,\ldots,n-1.$$

2.2 Reformulation into a stochastic shortest path problem

Since our state space is exponential in size, we cannot hope to solve our problem exactly using dynamic programming methods. All of the major methods for approximate dynamic programming consider infinite-horizon discounted cost problems. In order to use these methods for our finite-horizon stochastic scheduling problem, we recast our problem into a stochastic shortest path (SSP) problem. We refer to the following formulation of our problem as the *original SSP formulation.*

We introduce a terminating state $\bar{x}$ with the following property: only the states that have no more remaining jobs ($R_x = \emptyset$) can reach $\bar{x}$ in one step. Observe that for all states such that $R_x = \emptyset$ and at the terminating state $\bar{x}$, the set of actions available $\mathcal{A}_x$ is empty, and therefore the transition probabilities and time stage costs involving $\bar{x}$ are not affected by what action is taken. The transition probabilities involving $\bar{x}$ are

$$P_a(x,\bar{x}) = \begin{cases} 1 & \forall x \text{ such that } R_x = \emptyset \\ 1 & \text{if } x = \bar{x} \\ 0 & \text{otherwise} \end{cases}$$

and the time stage costs involving $\bar{x}$ are

$$g_a(x, \bar{x}) = \begin{cases} 0 & \forall x \text{ such that } R_x = \emptyset \\ 0 & \text{if } x = \bar{x}. \end{cases}$$

The cost-to-go function is therefore

$$J^*(x) = \min_u E\left[\sum_{t=0}^{T(x)} g_u(x_t, x_{t+1}) \middle| x_0 = x \right].$$

where $T(x)$ is the time stage when the system reaches the terminating state.

Recall that a stationary policy u is called a *proper* policy if, when using this policy, there is positive probability that the terminating state will be reached after a finite number of stages. Also recall that if all stationary policies are proper, then the cost-to-go function for the SSP problem is the unique solution to Bellman's equation [B01]. Since any policy in our scheduling problem requires one additional job to be scheduled at each time stage, we must always reach the terminating state with probability 1, regardless of the policy used. Therefore, to solve our problem exactly, we can solve Bellman's equation.

3 Solution method and bounds

We consider the linear programming approach to dynamic programming by de Farias and Van Roy [dV03] as a solution method to our problem. We briefly review the method and related results.

3.1 The linear programming approach to dynamic programming

Define the operator

$$TJ = \min_u \{g_u + \alpha P_u J\}.$$

where the minimization is carried out component-wise. We want to determine $J^{*,\alpha}$, the unique solution to Bellman's equation, $TJ = J$. The *exact linear programming (ELP)* approach to solving Bellman's equation solves the following optimization problem for any $c > 0$:

$$\begin{array}{ll} \text{maximize} & c^T J \\ \text{subject to} & TJ \geq J. \end{array}$$

Note that this optimization problem can be recast as the following linear program:

$$\begin{array}{ll} \text{maximize} & c^T J \\ \text{subject to} & g_a(x) + \alpha \sum_{y \in \mathcal{S}} P_a(x, y) J(y) \geq J(y), \quad \forall x \in \mathcal{S}, a \in \mathcal{A}_x. \end{array}$$

The *approximate linear programming (ALP)* approach to dynamic programming reduces the number of variables in the exact linear programming by the use of basis functions. Given a set of basis functions $\phi_1, \ldots, \phi_K$, we define the matrix

$$\Phi = \begin{bmatrix} | & & | \\ \phi_1 & \cdots & \phi_K \\ | & & | \end{bmatrix},$$

and in the hopes of computing a vector $\tilde{r}$ such that $\Phi\tilde{r}$ is a close approximation to $J^{*,\alpha}$, we solve the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & c^T \Phi r \\ \text{subject to} \quad & T\Phi r \geq \Phi r. \end{aligned} \tag{1}$$

This problem can be recast as a linear program just as in the ELP approach. Given a solution $\tilde{r}$, we hopefully can obtain a good policy by using the greedy policy with respect to $\Phi\tilde{r}$,

$$u(x) = \arg\min_{a \in \mathcal{A}_x} \left\{ g_a(x) + \alpha \sum_{y \in \mathcal{S}} P_a(x,y) (\Phi\tilde{r})(y) \right\}.$$

de Farias and Van Roy [dV03] proved the following bound on the error of the ALP solution.

Theorem 3.1. *(de Farias and Van Roy 2003) Let $\tilde{r}$ be a solution of the approximate LP (1), and $\alpha \in (0,1)$. Then, for any $v \in \mathbb{R}^K$ such that $(\Phi v)(x) > 0$ for all $x \in \mathcal{S}$ and $\alpha H \Phi v < \Phi v$,*

$$\|J^{*,\alpha} - \Phi\tilde{r}\|_{1,c} \leq \frac{2c^T \Phi v}{1 - \beta_{\Phi v}} \min_r \|J^{*,\alpha} - \Phi r\|_{\infty, 1/\Phi v}$$

where

$$(H\Phi v)(x) = \max_{a \in \mathcal{A}_x} \left\{ \sum_{y \in \mathcal{S}} P_a(x,y) (\Phi v)(y) \right\}$$

and

$$\beta_{\Phi v} = \max_x \frac{\alpha (H\Phi v)(x)}{(\Phi v)(x)}.$$

3.2 A uniform bound over number of jobs

We relax our stochastic shortest path problem by introducing discount factors $\alpha \in (0,1)$ at each time stage. We refer to this formulation as the *α-relaxed SSP formulation*. The cost-to-go function for this problem is

$$J^{*,\alpha}(x) = \min_u E\left[\sum_{t=0}^{T(x)} \alpha^t g_u(x_t, x_{t+1}) \middle| x_0 = x \right].$$

For this relaxation, we show that the error of the approximate linear programming solution is uniformly bounded over the number of jobs to be scheduled.

Theorem 3.2. *Assume that the holding cost $h(S)$ is bounded above by M for all subsets S of N. Let $\tilde{r}$ be the ALP solution to the α-relaxed SSP formulation of the stochastic scheduling problem. For $\alpha \in (0,1)$,*

$$\left\| J^{*,\alpha} - \Phi\tilde{r} \right\|_{1,c} \leq \frac{2M \max_{i \in N} E[p_i]}{1-\alpha}.$$

Proof. Due to the nature of our problem, we know that for any state x, $T(x) = |R_x| + 1$. However, since $g_a(x, \bar{x}) = 0$ for all x such that $R_x = \emptyset$, and $g_a(\bar{x}, \bar{x}) = 0$, when computing the cost-to-go function at state x, we only need to consider time stage costs for time stages 0 through $|R_x| - 1$. Since the holding cost h is bounded by above by M for all subsets of N, for some policy u we have

$$\begin{aligned}
J^{*,\alpha}(x) &= \min_u E\left[\sum_{t=0}^{|R_x|-1} \alpha^t g_{u(x_t)}(x_t, x_{t+1}) \middle| x_0 = x \right] \\
&\leq E\left[\sum_{t=0}^{|R_x|-1} \alpha^t g_{u(x_t)}(x_t, x_{t+1}) \middle| x_0 = x \right] \\
&\leq E\left[\sum_{t=0}^{|R_x|-1} g_{u(x_t)}(x_t, x_{t+1}) \middle| x_0 = x \right] \\
&= E\left[\sum_{t=0}^{|R_x|-1} \frac{1}{n} h(R_{x_t}) (C_{max}(x_{t+1}) - C_{max}(x_t)) \middle| x_0 = x, u \right] \\
&= E\left[\sum_{t=0}^{|R_x|-1} \frac{1}{n} h(R_{x_t}) p_{u(x_t)} \middle| x_0 = x \right] \\
&\leq E\left[\sum_{t=0}^{|R_x|-1} \frac{1}{n} M p_{u(x_t)} \middle| x_0 = x \right] \\
&= \frac{M}{n} \sum_{i \in R_x} E[p_i]
\end{aligned}$$

where $p_{u(x_t)}$ is the processing time of the job chosen at state x_t.

Consider the function

$$V(x) = \frac{k}{n} \sum_{i \in R_x} E[p_i]$$

for some constant k. Then for all $x \in \mathcal{S}$

$$\begin{aligned}
\alpha(HV)(x) &= \alpha \max_{a \in \mathcal{A}_x} \sum_{y \in \mathcal{S}} P_a(x,y) V(y) \\
&= \alpha k \max_{a \in \mathcal{A}_x} \sum_{y \in \mathcal{S}} P_a(x,y) \left(\frac{1}{n} \sum_{i \in R_y} E[p_i] \right)
\end{aligned}$$

$$
\begin{aligned}
&= \alpha \left(\frac{k}{n} \sum_{i \in R_x} E\left[p_i\right] - \frac{k}{n} E\left[p_{a^*}\right] \right) \\
&= V\left(x\right) \left(\alpha - \frac{\alpha E\left[p_{a^*}\right]}{\sum_{i \in R_x} E\left[p_i\right]} \right) \\
&< V\left(x\right)
\end{aligned}
$$

where $a^* \in \arg\max_{a \in \mathcal{A}_x} \sum P_a\left(x, y\right) V\left(y\right)$. Therefore, V is a Lyapunov function, and there is a $\beta < 1$ independent of the number of jobs n such that $\alpha H V \leq \beta V$. Also note that

$$
\begin{aligned}
\min_r \left\| J^{*,\alpha} - \Phi r \right\|_{\infty, 1/V} &\leq \left\| J^{*,\alpha} \right\|_{\infty, 1/V} \\
&= \max_{x \in \mathcal{S}} \frac{\left| J^{*,\alpha}\left(x\right) \right|}{V\left(x\right)} \\
&\leq \max_{x \in \mathcal{S}} \frac{\frac{M}{n} E\left[\sum_{i \in R_x} p_i\right]}{\frac{k}{n}\left[\sum_{i \in R_x} p_i\right]} \\
&= \frac{M}{k}
\end{aligned}
$$

This is uniformly bounded over the number of states.

Finally, we consider $c^T V$. Let c be some probability distribution such that $c\left(x\right) > 0$ for all $x \in \mathcal{S}$.

$$
\begin{aligned}
\sum_{x \in \mathcal{S}} c\left(x\right) V\left(x\right) &= \sum_{x \in S} c\left(x\right) \left(\frac{k}{n} \sum_{i \in R_x} E\left[p_i\right] \right) \\
&= k \sum_{x \in S} c\left(x\right) \left(\frac{1}{n} \sum_{i \in R_x} E\left[p_i\right] \right) \\
&\leq k \sum_{x \in S} c\left(x\right) \left(\max_{i \in N} E\left[p_i\right] \right) \\
&= k \max_{i \in N} E\left[p_i\right].
\end{aligned}
$$

This is uniformly bounded over the number of jobs, provided that the expected processing times of the jobs are bounded. Therefore, by Theorem 3.1, provided that $\sum_{i \in R_x} E\left[p_i\right]$ is one of our basis functions, we have that

$$
\begin{aligned}
\left\| J^{*,\alpha} - \Phi \tilde{r} \right\|_{1,c} &\leq \frac{2 c^T \Phi v}{1 - \beta_{\Phi v}} \min_r \left\| J^*_\alpha - \Phi r \right\|_{\infty, 1/\Phi v} \\
&\leq \frac{2M \max_{i \in N} E\left[p_i\right]}{1 - \alpha}
\end{aligned}
$$

which is uniformly bounded over the number of jobs, n. □

3.3 What happens when $\alpha = 1$?

Unfortunately, the bound in Theorem 3.2 explodes for $\alpha = 1$, and therefore does not directly apply to the original SSP formulation. The problem lies in the choice of the Lyapunov function. If we use the same Lyapunov function as in the proof of Theorem 3.2, we are left with the following equality

$$(HV)(x) = V(x)\left(1 - \frac{E[p_{a^*}]}{\sum_{i \in R_x} E[p_i]}\right).$$

As the number of jobs $n \to \infty$, the ratio of the expected processing time of any one job to the sum of the expected processing times of all jobs goes to zero. Therefore, by using the methods from the proof of Theorem 3.2, we cannot find a $\beta < 1$ such that $HV \leq \beta V$ regardless of the number of jobs.

However, since the ALP solution for the α-relaxed formulation is not that far away from the optimal solution to the α-relaxed formulation, we can show that the ALP solution obtained for the α-relaxed SSP formulation is also not that far away from the optimal solution of the original SSP formulation, depending on the value of α. First, we briefly present a well-known result from dynamic programming that relates the costs of the infinite-horizon discounted cost and average-cost problems.

Lemma 3.1. *For any stationary policy u and $\alpha \in (0, 1)$, we have*

$$J_{\alpha,u} = \frac{J_u}{1-\alpha} + h_u + O(|1-\alpha|)$$

where

$$J_{\alpha,u} = \sum_{k=0}^{\infty} \alpha^k P_u^k g_u, \qquad J_u = \left(\lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} P_u^k\right) g_u,$$

h_u is a vector satisfying

$$J_u + h_u = g_u + P_u h_u,$$

and $O(|1-\alpha|)$ is an α-dependent vector such that

$$\lim_{\alpha \to 1} O(|1-\alpha|) = 0.$$

For the stochastic scheduling problem formulated as a stochastic shortest path problem, the J_u in Lemma 3.1 is equal to zero. Recall that J^* is the cost-to-go function for the original SSP formulation. Therefore, Lemma 3.1 implies

$$\begin{aligned} J^* - J^{*,\alpha} &\leq J - J^{*,\alpha} \\ &= -O(|1-\alpha|) \\ \Rightarrow |J^* - J^{*,\alpha}| &\leq -O(|1-\alpha|) \end{aligned}$$

where J is the cost of using the optimal greedy policy for the α-relaxed SSP formulation in the original SSP formulation (h_u in Lemma 3.1). Since the cost-to-go functions for the original and α-relaxed SSP formulations are not that far

apart when α is large, the ALP solution for the α-relaxed SSP formulation is not that far away from the optimal solution to the original SSP formulation when α is large.

Corollary 3.1. *Let $\tilde{r}$ be the optimal solution to the ALP for the α-relaxed SSP formulation. Then for $\alpha \in (0,1)$,*

$$\|J^* - \Phi\tilde{r}\|_{1,c} \leq \frac{2M \max_{i \in N} E[p_i]}{1-\alpha} - c^T O(|1-\alpha|).$$

Proof. The result follows immediately from Theorem 3.2 and the arguments above:

$$\begin{aligned}
\|J^* - \Phi\tilde{r}\|_{1,c} &\leq \|J^* - J^{*,\alpha}\|_{1,c} + \|J^{*,\alpha} - \Phi\tilde{r}\|_{1,c} \\
&\leq c^T |J^* - J^{*,\alpha}| + \frac{2H \max_{i \in N} E[p_i]}{1-\alpha} \\
&\leq \frac{2H \max_{i \in N} E[p_i]}{1-\alpha} - c^T O(|1-\alpha|)
\end{aligned}$$

□

Although we have had difficulty in obtaining an error bound uniform over the number of jobs for the ALP solution when $\alpha = 1$, recent work [d04] indicates that relaxing the problem to include discount factors is the right approach to obtaining good bounds for the original SSP formulation.

4 Conclusion

We have shown that in theory, the approximate linear programming approach to dynamic programming should be a good solution method for the stochastic scheduling problem we studied. By relaxing the problem to include a discounting factor at each time stage, we obtained a bound on the error of the ALP solution that is uniform over the size of the problem. Using well-known results in dynamic programming, we also showed that by solving the relaxed problem, we obtain a solution that is not that far away from the optimal solution to the original problem.

The analysis above also provides a error bound uniform over the number of jobs for the multiple machine case. Although the state and action spaces of the problem with multiple machines are very complex, an upper bound on the cost-to-go function at any state can always be obtained by considering the cost-to-go function on just one machine.

One immediate avenue of further research would be to determine an error bound that is uniform over the number of jobs for the ALP solution to the original SSP formulation. Although our error bound does not grow as the size of the problem grows, the error bound is still potentially very large. It would be nice to see if a tighter bound on the errors can be obtained. Another direction of investigation would be to run computational experiments on using ALP

for our stochastic scheduling problem. It would be nice to see how the ALP approach performs in practice, and whether or not in reality ALP solves the original SSP formulation (with $\alpha = 1$) poorly as the size of the problem grows. Finally, the ALP method relies on solving a linear program with a constraint for every state-action pair, which in our problem would result in an extremely large optimization problem. One might investigate how the constraint sampling LP approach to dynamic programming [dV01] performs with the stochastic scheduling problem we studied, both theoretically and practically.

References

[B01] Bertsekas, D. (2001). *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont MA.

[BC99] Bertsekas, D., D. Castañon (1999). Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics* 5, pp. 89-108.

[d04] de Farias, D. P. (2004). Private communication, May 10, 2004.

[dV03] de Farias, D. P., B. Van Roy (2003). The linear programming approach to approximate dynamic programming. *Operations Research* 51, pp. 850-865.

[dV01] de Farias, D. P., B. Van Roy (2001). On constraint sampling for the linear programming approach to dynamic programming. *Mathematics of Operations Research*, forthcoming.

[KSW98] Karger, D., C. Stein, J. Wein (1998). Scheduling algorithms. *CRC Algorithms and Theory of Computation Handbook*, Chapter 35.

[MRW84] Möhring, R. H., F. J. Radermacher, G. Weiss (1984). Stochastic scheduling problems I: general strategies. *Zeitschrift für Operations Research* 28, pp. 193-260.

[MRW85] Möhring, R. H., F. J. Radermacher, G. Weiss (1985). Stochastic scheduling problems II: set strategies. *Zeitschrift für Operations Research* 29, pp. 65-104.

[MSU99] Möhring, R. H., A. S. Schulz, M. Uetz (1999). Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM* 46, pp. 924-942.

[R66] Rothkopf, M. H. (1966). Scheduling with random service times. *Management Science* 12, pp. 703-713.

[U96] Uetz, M. (1996). Algorithms for deterministic and stochastic scheduling. Ph.D. dissertation, Institut für Mathematik, Technische Universität Berlin, Germany.