Transportation Research Part A 46 (2012) 1216–1226

Contents lists available at SciVerse ScienceDirect

Transportation Research Part A

journal homepage: www.elsevier.com/locate/tra

Developing green fleet management strategies: Repair/retrofit/replacement decisions under environmental regulation

Timon H. Stasko *, H. Oliver Gao

Cornell University, Hollister Hall, Ithaca, NY 14853, USA

ARTICLE INFO

Article history:
Received 7 March 2011
Received in revised form 23 April 2012
Accepted 16 May 2012

Keywords:
Fleet management
Parallel asset replacement
Vehicle replacement
Emissions regulations
Approximate dynamic programming

ABSTRACT

The considerable cost of maintaining large fleets has generated interest in cost minimization strategies. With many related decisions, numerous constraints, and significant sources of uncertainty (e.g. vehicle breakdowns), fleet managers face complex dynamic optimization problems. Existing methodologies frequently make simplifying assumptions or fail to converge quickly for large problems. This paper presents an approximate dynamic programming approach for making vehicle purchase, resale, and retrofit decisions in a fleet setting with stochastic vehicle breakdowns. Value iteration is informed by dual variables from linear programs, as well as other bounds on vehicle shadow prices. Sample problems are based on a government fleet seeking to comply with emissions regulation. The model predicts the expected cost of compliance, the rules the fleet manager will use in deciding how to comply, and the regulation's impact on the value of vehicles in the fleet. Stricter regulation lowers the value of some vehicle categories while raising the value of others. Such insights can help guide regulators, as well as the fleet managers they oversee. The methodologies developed could be applied more broadly to general multi-asset replacement problems, many of which have similar structures.

1. Introduction

Many organizations, from private corporations to government agencies, depend on large fleets of vehicles to accomplish their objectives. Such large fleets require sizable capital investments and operational expenses. There is considerable interest in cost-minimizing vehicle replacement strategies, with an increasing emphasis on emissions reduction. Deciding when vehicles should be bought, sold, repaired, and retrofitted is no simple task, especially given the uncertainty surrounding future breakdowns. Numerous papers in the transportation research literature have provided valuable insights. Nonetheless, the models generally suffer from at least one of the following limitations: (1) simplifying assumptions prevent the model from being applied in many situations, and/or (2) the model has poor computational scalability, meaning it cannot always provide good recommendations in a reasonable time frame.

Numerous authors developed methods for computing "repair limits" (Drinkwater and Hastings, 1967; Ghellinck and Eppen, 1967; Hastings, 1968). The idea is that if a damaged asset requires more than this limit to repair, it should be replaced. Otherwise, it should be repaired and kept. Repair limits have been applied to multiple types of assets, including vehicles (Drinkwater and Hastings, 1967). Various authors have extended the vehicle or engine replacement problem by examining factors such as the loss of goodwill with riders due to breakdowns (Rust, 1987) and decreasing vehicle usage with age (Redmer, 2009). Unfortunately, models which consider only a single vehicle can make unimplementable

* Corresponding author. Tel.: +1 607 254 8334; fax: +1 607 255 9004.
E-mail addresses: ths9@cornell.edu (T.H. Stasko), hg55@cornell.edu (H.O. Gao).

0965-8564/$ - see front matter
http://dx.doi.org/10.1016/j.tra.2012.05.012

recommendations when applied to an entire fleet. The recommendation to replace all vehicles in a given category, for example, may violate budget constraints.

Replacement decisions in a fleet context can be complicated by a number of factors. Decisions can be linked by economies of scale in purchases, as well as a common budget. Furthermore, vehicles may be bought or sold without a ''replacement'' taking place if the size of the fleet is changing. Simms et al. (1984) tackled fleet vehicle replacement in a deterministic setting. Their non-linear objective, integer variables, and non-convex feasible region led them to a dynamic programming approach. Karabakal et al. (1994) presented an alternate methodology for replacing multiple assets under shared budgets, also in a deterministic setting. They used a branch-and-bound algorithm to solve an integer program, including a Lagrangian relaxation of budget constraints. Stasko and Gao (2010) developed an alternative integer program for optimizing fleet replacement strategies under budgets. The model added the capability of conducting retrofits, and of assigning value to emissions, but it did so in a purely deterministic setting. Suzuki and Pautsch (2005) created an integer program to solve for optimal fleet replacement strategies, and added a sensitivity analysis to help address uncertainty in some problem parameters. In particular, they examined the impacts of a range of percent increases and decreases in resale values and insurance premiums. The run times of their model (often over 6 h) made it necessary to accept non-integer solutions for the hundreds of runs performed in the sensitivity analysis. Furthermore, they were only able to explore sensitivity to two factors. More complicated forms of uncertainty, such as vehicle breakdowns, would be much more time intensive to model.

Enormous state spaces can make multi-asset replacement problems extremely difficult to solve, even without stochastic maintenance and repair costs. Numerous researchers have sought to reduce the size of the state space by making simplifying assumptions. Jones et al. (1991) demonstrated that the size of the problem could be dramatically reduced using two theorems. The first, known as the no-splitting rule, states that there is an optimal strategy in which all assets of the same age are treated the same way in any given period. The second, known as the older cluster replacement rule, states that it is only optimal to replace an asset if all older assets have been replaced. Childress and Durango-Cohen (2005) extended adaptations of these rules to the stochastic case, given comparable assumptions. The older cluster replacement rule requires several assumptions about cost structure. In particular, the assumption that the sum of maintenance cost, operating cost and salvage value is non-decreasing with respect to vehicle age is questionable, as relatively new vehicles often exhibit rapid declines in salvage value (McClurg and Chand, 2002). While the no-splitting rule does not require such assumptions about the cost structure, it falls apart in the presence of a binding budget constraint. Nonetheless, a significant portion of the literature on multi-asset replacement uses these or similar assumptions to reduce problem size (Jones et al., 1991; Chen, 1998; Jin and Kite-Powell, 2000; McClurg and Chand, 2002; Childress and Durango-Cohen, 2005).

Instead of limiting applications to small fleets (or single vehicles), assuming deterministic repair costs and vehicle lifespans, or depending on strong cost structure assumptions to shrink the problem, this paper presents a customized stochastic approximate dynamic program (ADP) which is well suited to dealing with large state spaces. The new model is designed to merge the capabilities of several previous models, while maintaining tractability by taking advantage of the strengths of simulation, linear programming, and dynamic programming.

The presented formulation develops vehicle purchase, resale, and retrofit policies, given stochastic maintenance and repair costs and vehicle failures. The objective is to minimize expected discounted net costs. In each period, there must be enough vehicles to meet a deterministic demand, while complying with environmental regulations. The recommended policies are described by integer programs which are adapted to the current makeup of the fleet, meaning that the policies do not have to be entirely recomputed every time a vehicle breaks down earlier or later than expected.

Such a tool could be of value to fleet managers whose situations could not be accurately captured by previous narrower models. It could also be of use to regulators seeking to better understand the impacts of potential regulation. The inclusion of retrofits and environmental regulations is particularly timely given the diesel emission laws passed in New York and California which require many government departments to retrofit or replace portions of their fleet before a series of deadlines (NYS DEC, 2009; CARB, 2006). Further related regulation which would apply to private fleets is being considered (CARB, 2011). The New York laws are the inspiration for sample problems. As part of the adaptive strategy, the ADP provides estimates of vehicle values. This feature can be used to reveal how potential regulatory requirements would impact the value of a fleet.

The combined problem of determining purchase, resale, and retrofit policies is referred to as the fleet upkeep problem. Section 2 describes the ADP approach and explains the reasoning behind it, while Section 3 presents a sample implementation and interprets the results. Section 4 summarizes the findings and potential research directions.

2. Methodology

2.1. Selecting a value iteration ADP approach

Stochastic dynamic programming is well equipped to represent the fleet upkeep problem for a number of reasons. Stochastic dynamic programming can handle the discrete nature of vehicles and accurately represent the dynamic interaction between stochastic breakdown events and fleet owner decisions. Breakdown events occur randomly, and fleet owners respond (possibly by repairing the vehicle, or replacing it). Future breakdown events then depend on the actions taken by the fleet owner in response to earlier breakdown events. This feedback effect makes it challenging to pregenerate scenarios, which are commonly used in other stochastic optimization techniques.

Furthermore, stochastic dynamic programming is capable of representing the fleet manager's changing access to information over time. When making decisions at a given point in time, the fleet manager knows the current state of the fleet, as well as something about the likelihood of future breakdown events. This continuously changing knowledge is easily captured by a stochastic dynamic programming framework.

A stochastic version of Bellman's equation is given by expression (1) where $V_t(S_t)$ is the value of being in state S_t at the start of period t (assuming optimal behavior), $N_t(S_t,x_t)$ is the net benefit experienced at the start of period t when taking action x_t in state S_t, and ρ is the discount factor.

$$V_t(S_t) = max_{x_t}\{N_t(S_t, x_t) + \rho E[V_{t+1}(S_{t+1}(S_t, x_t))]\} \tag{1}$$

In the fleet upkeep problem, the state space describes the set of possible conditions the fleet could be in at any given point in time. The state of the fleet is defined by a set of integer variables f_{ajk}, each indicating how many vehicles exist in a relevant category. A category is defined by vehicle age a, maintenance status j, and retrofit status k. Actions described by x_t are vehicle purchases, sales, repairs, and retrofits. More precisely, x_t is a vector consisting of the decision variables outlined in Section 2.2.3. N_t is the vehicle sales revenue minus the costs due to other actions. Uncertainty stems from the fact that future maintenance statuses and vehicle failures are not known. Thus, the expectation is taken over possible vehicle maintenance status and vehicle failure combinations. For the purposes of this paper, the maintenance status of a vehicle indicates how much money must be spent on maintenance and repairs in order to keep using that vehicle.

Once the fleet upkeep problem is framed as a stochastic dynamic program, the next step is to select a method for solving the program. It is well known that dynamic programs grow quickly with the dimension of the state space. This is often referred to as dynamic programming's "curse of dimensionality." A relatively simple example might have 25 possible ages, three maintenance statuses, and two retrofit statuses (compliant and non-compliant). The state space could be represented using 150 variables, one for each vehicle category. Even if no category ever has more than 19 vehicles, there are a whopping 20^{150} possible states of the fleet.

In the example problems described in Section 3, the average time to compute the value of a state was at least 0.01 s. At this pace, it would take roughly $4.5 * 10^{185}$ years to evaluate the value of every state for a single period, which is significantly longer than the estimated of the age of the universe (NASA, 2009). Traditional backwards dynamic programming requires computing the value of every such state in every time period, an obviously infeasible task. Alternatively, dynamic programs can be reformulated as linear programs with variables for each state and constraints for each state-action combination (Bertsekas, 1987). Given the vast size the of the state and action spaces, however, this approach also has limited applicability.

Approximate dynamic programming techniques are often able to produce high quality solutions, despite examining only a small fraction of possible states. Among ADP approaches, value iteration is particularly well suited for the fleet upkeep problem, because retrofit constraints will generally change over time. Policy iteration, an alternative, is popular for steady-state infinite-horizon problems (Powell, 2007).

An outline of the value iteration approach employed is provided in Table 1. Forward passes through time act as sequences of simulation steps and optimization steps, capturing random effects and acting in response to them. This approach allows the optimizer to focus on understanding regions of the state space which are of greatest importance, and to extrapolate based on the findings.

Developing an appropriate value function form is one of the key challenges when formulating an ADP. This paper employs a linear value function which assigns a value to each vehicle category, and allows these values to change over time (e.g. when regulatory mandates take effect). The value of the fleet is simply the sum of the values of the vehicles it contains. This functional form allows for subproblems to be solved efficiently, and the parameters that define it have intuitive meaning. This makes results easily interpretable, and facilitates identification of errors in implementation.

Table 1
ADP Algorithm Outline.

1. Initialize. Input data on current fleet status, future demands, and future retrofit regulation. Set period = 1
2. Solve single period IP using network flow LP formulation defined by expressions (2)–(6)
3. If period 2 or later:
   Update the value function approximation by using expressions (7) and (8)
   to adjust the value of each vehicle category in the previous period
4. If not the last period:
   a. Using the transition function described in Section 2.4, update fleet status based on manager actions (repairs, sales, purchases, retrofits), and then based on random breakdown events
   b. Move to the next time period. Update demand and retrofit requirements
   c. Go to step 2

   If the last period:
   a. Update the final fleet values to equal the average over the last year (assumed steady state)
   b. If not final iteration:
   Reset to initial fleet status and period 1 demand/retrofit requirements. Go to step 2
   c. If final iteration: Go to step 5
5. Compute performance metrics. Output results

2.2. Network flow formulation

Given expression (1), the next step is to determine how to solve for the optimal set of actions, x_t^*. Largely because of the form selected for the value function approximation, this problem can be modeled as an integer program (IP). The objective and constraints are all linear in the decision variables. This integer program effectively forms the policy used to make decisions at a given point in time.

IPs are NP-hard, and no polynomial time algorithm for solving them is known. There are well known polynomial time algorithms for solving linear programs (LPs) without integrality constraints, as well as a worst-case exponential algorithm (known as the simplex method) which works very well in practice (Kleinberg and Tardos, 2006). For this reason, it is natural to seek a linear program formulation which will yield integer solutions.

There are several classes of network flow problems for which the simplex algorithm produces integer solutions. The minimum cost flow problem is one such problem class (Sierksma, 1996), and it can be used to model the single period fleet upkeep problem. This is possible because the discrete nature of vehicles will yield only integer supplies, demands, and upper bounds on link flows. A network illustration of a simple single period fleet upkeep problem is presented in Fig. 1.

Vehicles in the initial fleet flow from source S1, while potential new purchases flow from source S2. All flows terminate at sink T. Costs on the edges produce the proper objective, and the capacity constraints combine with conservation of flow to construct the proper feasible region. For example, the capacity of the link from "Not Available" to T ensures there are enough vehicles available to meet demand. Extensions such as allowing purchases of non-compliant vehicles or multiple maintenance statuses complicate the picture, but they can still be represented as a network flow problem. A linear program which allows for such extensions is outlined below.

2.2.1. Sets

A	set of vehicle ages
J	set of maintenance statuses
K	set of retrofit statuses
T	set of time periods (assumed to have same resolution as vehicle ages)

2.2.2. LP parameters

f_{ajk}	number of age a vehicles in maintenance status j and retrofit status k at the start of the period
$c_{ajk_1k_2}$	cost of keeping an age a vehicle currently in maintenance status j and retrofit status k_1, to be put in new retrofit status k_2
p_k	price of a new vehicle in retrofit status k
u_k	maximum number of new vehicles in retrofit status k which can be purchased
r_{aj}	net resale revenue for a vehicle of age a in maintenance status j
v_{ak}	discounted future value of a kept vehicle of age a in retrofit status k
w_k	discounted future value of a bought vehicle in retrofit status k
ϕ	demand for vehicles in current period which must be met
ψ_k	maximum number of vehicles in retrofit state k to be held or bought

2.2.3. Decision variables

g_k	number of new vehicles in retrofit status k to be bought
$h_{ajk_1k_2}$	number of vehicles of age a vehicle currently in maintenance state j and retrofit status k_1, to be kept and put in new retrofit status k_2
q_{ajk}	number of vehicles of age a vehicle currently in maintenance state j and retrofit status k to be resold

2.2.4. Objective

$$max \sum_{k_1 \in K} \left\{ (w_{k_1} - p_{k_1}) g_{k_1} + \sum_{a \in A} \sum_{j \in J} \sum_{k_2 \in K} (v_{ak} - c_{ajk_1k_2}) h_{ajk_1k_2} \right\} + \sum_{a \in A} \sum_{j \in J} \sum_{k \in K} r_{aj} q_{ajk} \tag{2}$$

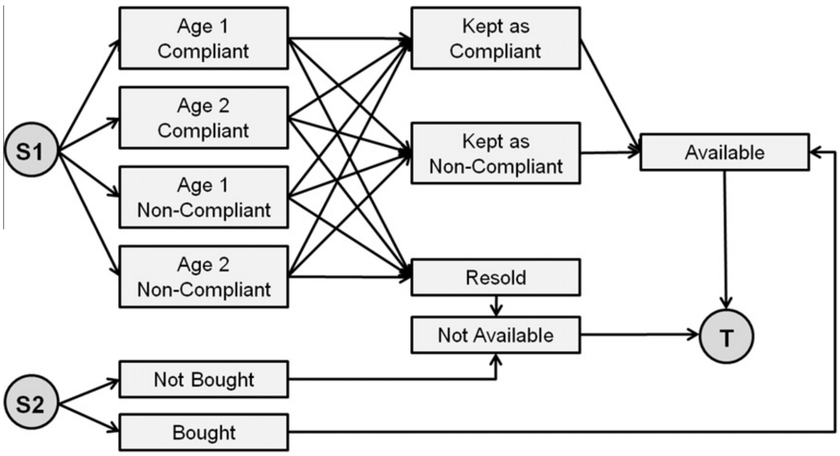

Fig. 1. Network flow representation of a simple single period fleet upkeep problem.

2.2.5. Constraints

$$q_{ajk_1} + \sum_{k_2 \in K} h_{ajk_1k_2} = f_{ajk_1} \quad \forall\ a \in A, j \in J, k_1 \in K \tag{3}$$

$$\left\{ \sum_{a \in A} \sum_{j \in J} \sum_{k_1 \in K} \sum_{k_2 \in K} h_{ajk_1k_2} \right\} + \sum_{k \in K} g_k \geqslant \phi \tag{4}$$

$$g_{k_2} + \sum_{a \in A} \sum_{j \in J} \sum_{k_1 \in K} h_{ajk_1k_2} \leqslant \psi_{k_2} \quad \forall\ k_2 \in K \tag{5}$$

$$g_k \leqslant u_k \quad \forall\ k \in K \tag{6}$$

The objective, given by expression (2), is to maximize the discounted future value of the fleet, plus vehicle sales revenue from the current period, minus costs from the current period (maintenance, retrofits, and purchases). Expression (3) is a constraint requiring conservation of flow for vehicles in the existing fleet. Expression (4) requires that there are enough vehicles to meet demands. Expression (5) caps the number of vehicles bought or kept in each retrofit status. Expression (6) caps vehicle purchases. This cap is used to create a network flow formulation, and is assumed to be high enough that the problem remains feasible.

2.3. Updating the value function

Once optimal actions are determined for the current period, the next step is to update the value function estimate. Vehicle shadow prices from the current period are used to update the vehicle value estimates used in the previous period's LP. Naturally, these improved value estimates would not be used until the next forward pass.

The value function in iteration n is defined by a set of parameters, v^n_{akt}, indicating the expected discounted future value of keeping a vehicle of age a, and retrofit status k, at the start of period t, as well as a set of parameters w^n_{kt}, indicating the expected discounted future value of a new vehicle in retrofit status k, bought at the start of period t. Because v^n_{akt} and w^n_{kt} are the means of a random variables, it does not make sense to ignore the previous estimates whenever new observations are found. Instead, the new estimates are weighted combinations of the old estimates and the discounted average of shadow prices from period $t+1$, as given by expressions (7) and (8). The shadow prices are averaged over the different maintenance statuses (including complete failure).

$$v^n_{akt} = (1-\alpha_{n-1})v^{n-1}_{akt} + \alpha_{n-1}\rho\left[\tau_{(a+1)}\varphi + \sum_{j \in J}\pi_{(a+1)j}\lambda^{n-1}_{(a+1)jk(t+1)}\right] \tag{7}$$

$$w^n_{kt} = (1-\alpha_{n-1})w^{n-1}_{kt} + \alpha_{n-1}\rho\left[\tau_1\varphi + \sum_{j \in J}\pi_{1j}\lambda^{n-1}_{1jk(t+1)}\right] \tag{8}$$

In expressions (7) and (8), φ is the resale revenue of a vehicle which has failed beyond repair, while τ_a is the probability of complete failure for a vehicle of age a, π_{aj} is the probability of being in maintenance state j for a vehicle of age a, and λ^n_{ajkt} is the shadow price for a vehicle of age a in maintenance status j and retrofit status k during period t of the nth iteration.

Selecting appropriate alpha values for step sizes is critically important. Both theory and experience can guide step size selection. Theory comes from conditions for convergence proofs of stochastic gradient algorithms, which essentially require that step sizes decline according to a harmonic sequence. Experience, on the other hand, indicates that a simple $\alpha_{n-1} = \frac{1}{n}$ step size rule drops too quickly (Powell, 2007). As a result, the current ADP implementation uses a well known step size rule which is based on the generalized harmonic sequence given in expression (9).

$$\alpha_{n-1} = \frac{\delta}{\delta + n - 1} \tag{9}$$

In order to implement expressions (7) and (8), it is necessary to more precisely define shadow prices, and develop a method for estimating them. In general, a shadow price is the rate of change in the optimal objective function with respect to change in the amount of one resource. A simple means of obtaining shadow prices is to add or subtract a unit of the resource in question, resolve the LP, and compare objective values. While reliable, this method can be very time consuming when many shadow prices are required.

In linear programming, dual variables are commonly used to determine shadow prices. Unfortunately, obtaining shadow prices is not always as simple as outputting the dual variables corresponding to the optimal solution. Classical linear programming texts have been criticized for misleading readers about the equivalence of shadow prices and dual variables (Akgül, 1984). The equation of dual variables and shadow prices is based on the assumption of non-degeneracy. If the optimal primal solution is degenerate, however, there may be alternative dual values, meaning that the shadow prices are no longer necessarily equal to the set of dual variables output by the solver (Lin, 2010). Even a simple fleet upkeep problem with only a few vehicles can exhibit primal degeneracy. This can cause non-compliant vehicles to be erroneously assigned the same shadow price as compliant vehicles, significantly impacting results.

The operations research community has been struggling to deal with shadow prices of degenerate LPs for some time. Various approaches require solving different, albeit smaller LPs for each shadow price sought (Akgül, 1984; Lin, 2010). Additionally, it has been proven that if we know the set of optimal dual solutions $y \in D^*$, then:

$$\lambda_z^+ = min\{y_z : y \in D^*\} \tag{10}$$

$$\lambda_z^- = max\{y_z : y \in D^*\} \tag{11}$$

where λ_z^+ is the shadow price for an additional unit of resource z, while λ_z^- is the shadow price of the last unit of resource z, and the primal problem is a maximization (Lin, 2010). Essentially, this means that the dual variable for a particular vehicle is an upper bound on the value of another such vehicle.

The ADP presented uses dual variables as upper bounds on shadow prices, and constructs lower bounds for comparison. Lower bounds are constructed by considering what could be done with the additional vehicle. If kept, the vehicle might be retrofitted, and it might eliminate the need for a new vehicle purchase, depending on which constraints are binding. Alternatively, the vehicle could be sold. If upper and lower bounds are sufficiently close (within $100 for sample problems), then the average of the bounds is used for the shadow price. Otherwise, the actual shadow price is determined by perturbing the right-hand-side vector and resolving the LP.

This hybrid approach to shadow price estimation proved far more accurate than depending on dual variables and far faster than perturbing the right-hand-side vector in every case. On relatively small sample problems, pure perturbations took more than 30 times longer than the hybrid approach, which only needed to perform perturbations roughly 1-10% of the time. Larger sample problems which used quarters as time periods instead of years could not be solved using pure perturbations in a reasonable time frame. When the program was stopped on the second day it was on pace to finish in roughly two weeks. When using the hybrid approach, the ADP converged in a few hours. Despite being relatively small, and having "warm starts," the perturbed LPs associated with each shadow price are time consuming to solve, and are better used as a last resort than as a standard approach.

2.4. Moving between states

Recall that each iteration of the ADP is a simulated forward pass through time. Once actions are determined for a given period, and the value function for the past period has been updated, the next step is to move forward to the next period. This is accomplished with the transition function. Using several nested loops, the transition function generates random variables describing breakdown events, and produces the pre-decision state of the fleet for the next time period. All vehicles are aged by one period. Some vehicles fail completely and are sold for scrap. Those remaining are randomly assigned a maintenance status according to the appropriate probabilities.

3. Illustrative examples

3.1. Situation description

The example problems are based on the situation facing public fleet managers when the Diesel Emission Reduction Act of 2006 was signed in New York. A large legacy dump truck fleet must continue to satisfy constant demand for 2000 vehicles.

By the start of 2009, only 1333 non-compliant vehicles can remain in the fleet. By the start of 2010, that number must be cut to 667 or below, and no non-compliant vehicles can remain after the start of 2011. All 2007 model year and newer vehicles are compliant. All pre-2007 vehicles are assumed to require a $15,000 filter to become compliant, regardless of age. In reality, vehicle and driving profile differences can cause technology requirements to vary, but using a single cost figure will make the results more easily interpretable. The model structure does allow retrofit costs to vary. As with any optimization of long-term fleet management, building the example problem requires the estimation of parameters, such as future vehicle prices. Those are not the focus of this paper, but previous work on this subject can be found in Gao and Stasko (2009).

3.2. Testing convergence in a deterministic case

It has been demonstrated that under a fairly broad set of conditions ADP algorithms do converge eventually (Powell, 2007). In order to test the speed of said convergence, a deterministic version of the above problem was constructed. This allows for the ADP's objective at each iteration to be compared to the (provably optimal) objective produced by a single large IP. In order to make this comparison, the objective function is changed slightly so that the fleet remaining at the end of the simulation is sold at exogenous market prices, instead of valued at endogenously determined prices. This preserves linearity in the IP, which makes it possible to use standard linear IP solvers.

Despite a deliberately naïve initial guess that all vehicles are worth $80,000, independent of age and condition, the ADP produced a solution within 1% of the CPLEX11.2.1 IP optimum by the 23rd iteration (taking about 1.5 h). The discounted net cost is plotted as a function of the iteration in Fig. 2a, with the IP optimum designated by a dashed line. In the absence of an IP optimum for comparison, several factors can offer clues that the ADP has reached an optimum. Perhaps the most obvious sign is a decrease in the rate of improvement of the objective function, as is clearly the case in Fig. 2a. Slowed improvement can be decieving however, and may not indicate optimality. It is possible that the step sizes have simply declined to the point where the value function is changing too slowly to noticeably improve the objective.

In order to avoid step size issues, one can directly compare shadow prices to the previous iteration's value function. Fig. 2b plots the maximum and average absolute differences between value estimates based on current shadow prices

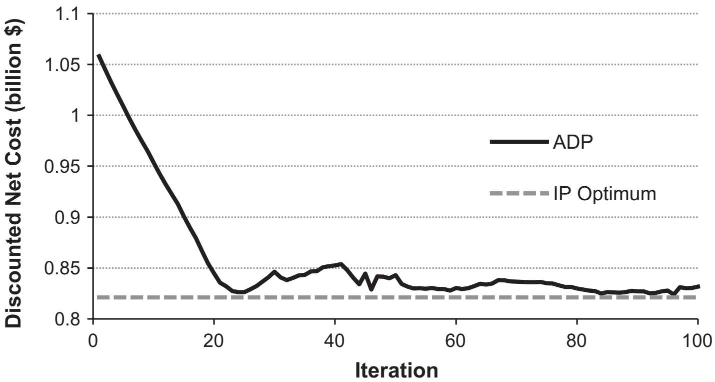

Fig. 2a. Objective function convergence in a deterministic case.

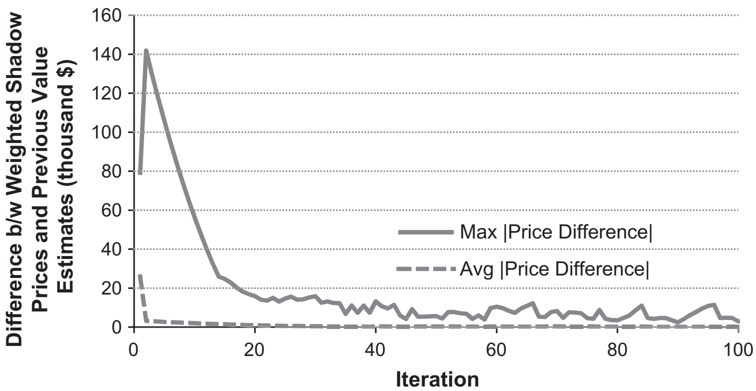

Fig. 2b. Shadow price convergence in a deterministic case.

and previous value estimates. At the start of the ADP, the average absolute difference is in the tens of thousands dollars, with the maximum absolute difference topping $100,000. By the 100th iteration, the average absolute difference is a little over $100, and the maximum is a few thousand dollars. The fact that the value function is not going to change dramatically, regardless of the step size, provides a helpful hint that the ADP has converged, but it does not equate to a guarantee.

In addition to allowing the use of a single IP, this simplified deterministic problem allows us to solve for steady-state vehicle values analytically. First, the optimal lifespan is computed by minimizing the equivalent uniform annual cost as in Newnan et al. (2002). Second, the value of having a vehicle at the end of the current period, which will be retired at that time, is set equal to the resale revenue discounted to the present, as in expression (12). Third, all younger vehicles have their values recursively calculated according to expression (13).

$$V[\theta_f] = \beta/(1+\delta) \tag{12}$$

$$V[\theta_f - \mu] = \frac{V[\theta_f - \mu + 1]}{1+\delta} + \frac{\left(L + \frac{L}{(1+\delta)^{\theta_f} - 1}\right)\left(1 - \frac{1}{1+\delta}\right)}{1+\delta} - \frac{m(\theta_f - \mu + 1)}{1+\delta} \tag{13}$$

where:

$V[\theta]$	is the value of a vehicle of age θ
θ_f	is the last age at which the vehicle is used
β	is the resale revenue
δ	is the discount rate
L	is the lifetime cost of a vehicle held for the optimal lifespan, discounted to the purchase date
$m[\theta]$	is the maintenance and repair cost paid when using a vehicle of age a

The vehicle value estimates produced by the ADP at each iteration were recorded and it became apparent that they quickly converged to the analytical solution. The mean absolute percentage error (MAPE) was computed for each iteration and plotted in Fig. 3. The initial guess, which is close to the average vehicle value, yielded a MAPE of roughly 37%. At first, estimates worsened, with the MAPE peaking at just over 64% in the second iteration. The ADP quickly recovered, however, improving to a MAPE just under 1% by the 20th iteration. By the end of the simulation, the MAPE was hovering between 0.1% and 0.3%.

The convergence of vehicle values remained relatively consistent as various parameters, such as scrap values, were changed. In one test, the initial value guess was set an order of magnitude too high, at $800,000 per vehicle, well above the $160,000 new vehicle purchase price. Convergence was noticeably slower, but the ADP had clearly managed to head in the right direction despite the very cold start. By the end of 250 iterations (taking 4.36 h), the R^2 was a respecTable 0.839.

3.3. Exploration of a stochastic case

In order to better represent reality and more fully illustrate some of the ADP's capabilities, a stochastic version of the problem was developed, including uncertain vehicle lifetimes and maintenance costs. Neither the single IP approach nor the analytical solution for vehicle values will work in the stochastic case, though the latter can be used to provide an initial guess for the value function. In the stochastic example, the expected vehicle lifetime and maintenance costs match those used in the deterministic case. As shown in Fig. 4, convergence follows a similar pattern to the deterministic case, though

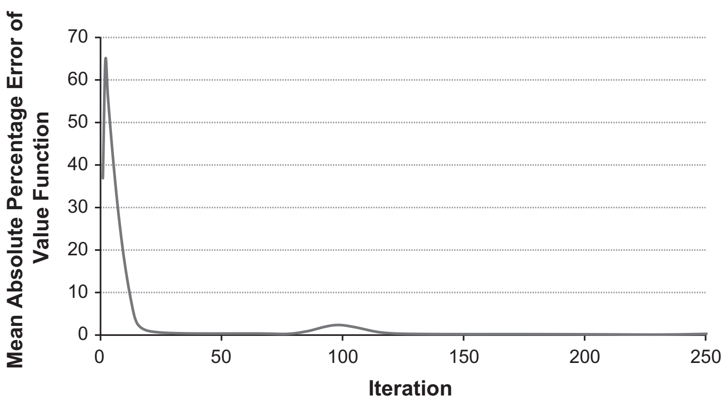

Fig. 3. Value function convergence in a deterministic case.

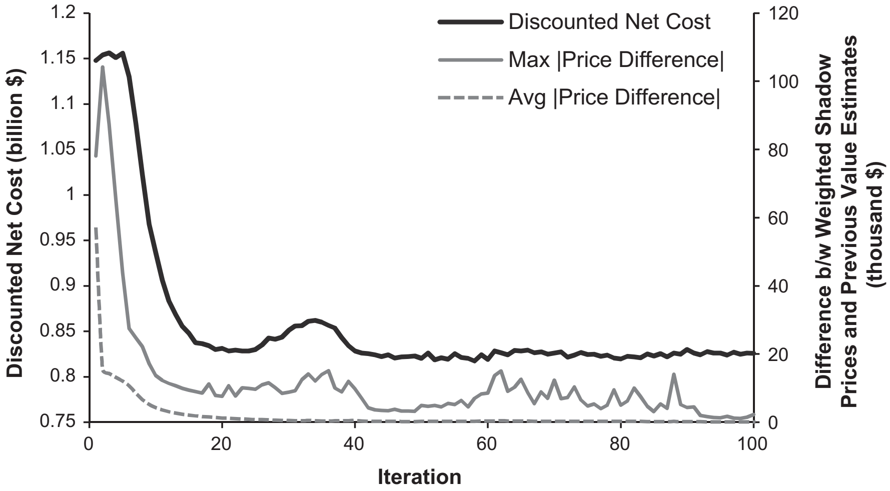

Fig. 4. Objective function and shadow price convergence in a stochastic case.

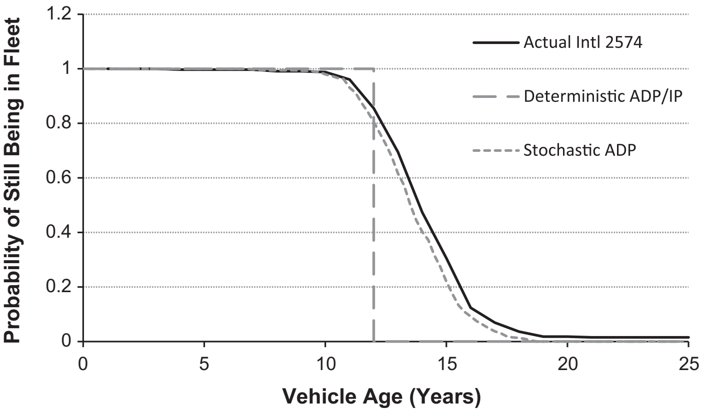

Fig. 5. Probability a given vehicle will remain in the fleet X years.

there is no IP optimum available for comparison. The ADP was further run to 350 iterations, yielding insignificant change in the objective function.

The degree to which stochastic maintenance costs enable the ADP to better represent reality is made clear by Fig. 5. It plots the probability that a vehicle will still be in the fleet at a range of ages (assuming a steady state without retrofit regulation). The deterministic model recommends replacing all vehicles at the same age. The solid curve, which is derived from the actual auction dates of 331 class 8 International 2574 dump trucks, indicates that vehicles are not in fact retired at a consistent age. In this case, they are phased out over a period of roughly five years. Most vehicles are replaced when they would need a major repair to remain operational. The stochastic ADP result strongly resembles the actual replacement pattern.

Depending on the makeup of a given fleet, it is possible for retrofit regulation to actually increase the value of a fleet, even if regulation increases the expected fleet upkeep cost. This is because regulation can effectively expose latent capabilities of some vehicles. Put another way, the cost of the required future work goes up, but so does the ability of the current fleet to offset some of those costs. Vehicle value estimates for compliant and non-compliant vehicles in the first period are plotted as a function of age in Fig. 6, along with vehicle values in the absence of retrofit regulation. As one would suspect, compliant and non-compliant vehicles have the same value in the absence of regulation.

When the regulation is imposed, two types of changes in vehicle values occur. First, relatively new non-compliant vehicles decrease in value. They will likely have to be retrofitted or retired earlier than previously planned, reducing their ability to prevent future costs. Second, compliant vehicles which are a few years old increase in value. These are vehicles which might previously have been retired during the years of the phase in, but they might be able to be kept slightly longer and prevent some of the more painful early retirements. New compliant vehicles do not change in value, because they were already very likely to be kept through the phase in.

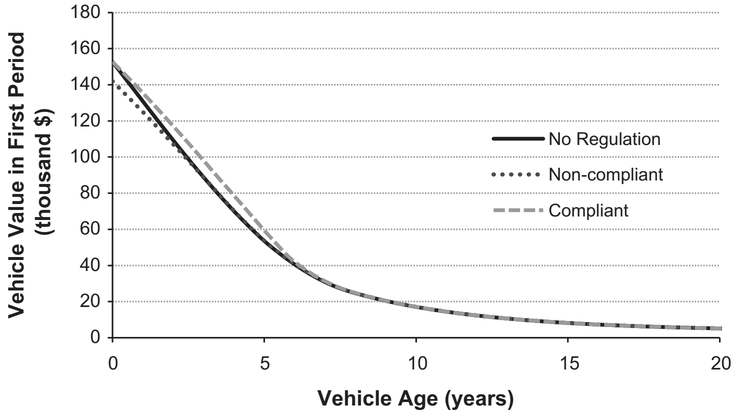

Fig. 6. Vehicle values in first time period.

4. Conclusion

It is possible to use an approximate dynamic program to model the fleet upkeep problem. The ADP presented converges in a reasonable time frame, even when given an intentionally poor initial value function estimate. In deterministic examples, the ADP came well within 1% of the IP optimum in a few dozen iterations and its value function approached the analytical solution. Unlike the IP and analytical solution, the ADP is able to handle the stochastic case just as easily. The stochastic case, which allows for a distribution of vehicle lifespans, far better represents reality.

The ADP presented uses a value function form which assigns a value to each vehicle category in each time period. Shadow prices are a powerful tool for informing the value function, but they can be computationally intensive to compute when perturbing the right-hand-side vector. The computational expense can be dramatically reduced in many cases by using bounds on shadow prices provided by dual variables and other outputs of the subproblem linear programs. The subproblems can be framed as network flow problems, allowing each to be solved by a single LP.

The ADP provides cost-minimizing policies in the form of integer programs which adapt to the current stochastic state of the fleet. These strategies can be much more sophisticated than traditional methods such as fixed retirement ages or repair limits, allowing fleet managers to better react to dynamic regulatory environments. In addition, the ADP outputs vehicle value estimates for all relevant vehicle categories. This can reveal counterintuitive ways in which retrofit or emission reduction regulation might alter the values of fleets or individual vehicles, thus creating distributional impacts.

Future work will seek to include new factors and apply the ADP presented to related problems. New factors could include fuel price volatility and regulatory uncertainty. The ADP could be adapted for use in a wide range of multiple asset retrofit/replacement problems, from overseeing the upkeep of industrial equipment to maintaining a portfolio of buildings. In these and many other settings, managers are faced with the challenge of determining when to buy, sell, repair, and upgrade their assets, given changing demands and uncertain performance.

Acknowledgments

The authors thank the US Environmental Protection Agency (EPA). This paper was developed under STAR Fellowship Assistance Agreement no. 91717401-0 awarded by the EPA. It has not been formally reviewed by EPA. The views expressed in this paper are solely those of the authors and EPA does not endorse any products or commercial services mentioned in this paper.

References

Akgül, M., 1984. A note on shadow prices in linear programming. The Journal of the Operational Research Society 35 (5), 425–431.

Bertsekas, D.P., 1987. Dynamic Programming: Deterministic and Stochastic Models. Prentice Hall International, London.

California Air Resources Board [CARB], 2006. Fact Sheet: Fleet Rule for Public Agencies and Utilities. <http://www.arb.ca.gov/msprog/publicfleets/publicfleetsfactsheet.pdf>.

California Air Resources Board [CARB], 2011. Truck and Bus Regulation: On-Road Heavy-Duty Diesel Vehicles (In-Use) Regulation. <http://www.arb.ca.gov/msprog/onrdiesel/onrdiesel.htm>.

Chen, Z., 1998. Solution algorithms for the parallel replacement problem under economy of scale. Naval Research Logistics 45 (3), 279–295.

Childress, S., Durango-Cohen, P., 2005. On parallel machine replacement problems with general replacement cost functions and stochastic deterioration. Naval Research Logistics 52 (5), 409–419.

Drinkwater, R.W., Hastings, N.A.J., 1967. An economic replacement model. Operational Research Quarterly 18 (2), 121–138.

Gao, H.O., Stasko, T.H., 2009. Diversification in the driveway: mean-variance optimization for greenhouse gas emissions reduction from the next generation of vehicles. Energy Policy 37 (12), 50195027.

Ghellinck, G.T., Eppen, G.D., 1967. Linear programming solutions for separable Markovian decision problems. Management Science 13 (5), 371–394.

Hastings, N.A.J., 1968. Some notes on dynamic programming and replacement. Operational Research Quarterly 19 (4), 453–464.
Jin, D., Kite-Powell, H.L., 2000. Optimal fleet utilization and replacement. Transportation Research Part E 36 (1), 3–20.
Jones, P.C., Zydiak, J.L., Hopp, W.J., 1991. Parallel machine replacement. Naval Research Logistics 38 (3), 351–365.
Karabakal, N., Lohmann, J.R., Bean, J.C., 1994. Parallel replacement under capital rationing constraints. Management Science 40 (3), 305–319.
Kleinberg, J., Tardos, É., 2006. Algorithm Design. Pearson Addison Wesley, Boston.
Lin, C., 2010. Computing shadow prices/costs of degenerate LP problems with reduced simplex tables. Expert Systems with Applications 37 (8), 5848–5855.
McClurg, T., Chand, S., 2002. A parallel machine replacement model. Naval Research Logistics 49 (3), 275–287.
National Aeronautics and Space Administration [NASA], 2009. How old is the universe? <http://map.gsfc.nasa.gov/universe/uni_age.html>.
New York State Department of Environmental Conservation [NYS DEC], 2009. Adopted Part 248 – Use of Ultra Low Sulfur Diesel Fuel and Best Available Retrofit Technology for Heavy Duty Vehicles, and Part 200 – General Provisions. <http://www.dec.ny.gov/regulations/56126.html>.
Newnan, D.G., Lavelle, J.P., Eschenbach, T.G., 2002. Essentials of Engineering Economic Analysis, second ed. Oxford University Press, New York.
Powell, W.B., 2007. Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & Sons, Hoboken, NJ.
Redmer, A., 2009. Optimisation of the exploitation period of individual vehicles in freight transportation companies. Transportation Research Part E 45 (6), 978–987.
Rust, J., 1987. Optimal replacement of GMC bus engines: an empirical model of Harold Zurcher. Econometrica 55 (5), 999–1033.
Sierksma, G., 1996. Linear and Integer Programming: Theory and Practice. Marcel Dekker, New York.
Suzuki, Y., Pautsch, G.R., 2005. A vehicle replacement policy for motor carriers in an unsteady economy. Transportation Research Part A 39 (5), 463–480.
Simms, B.W., Lamarre, B.G., Jardine, A.K.S., Boudreau, A., 1984. Optimal buy, operate, and sell policies for fleets of vehicles. European Journal of Operational Research 15 (2), 183–195.
Stasko, T.H., Gao, H.O., 2010. Reducing transit fleet emissions through vehicle retrofits, replacements, and usage changes over multiple time periods. Transportation Research Part D 15 (5), 254–262.