IIE Transactions (2006) **38**, 159–172
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170500333384

Combining cost-based and rule-based knowledge in complex resource allocation problems

ARUN MARAR[1], WARREN B. POWELL[1] and SANJEEV KULKARNI[2]

[1]*Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544*
[2]*Department of Electrical Engineering, Princeton University, Princeton, NJ 08544*

Downloaded by [Princeton University] at 07:52 04 January 2013

A major challenge in the formulation of optimization models for large-scale, complex operational problems is that some data are impossible or uneconomical to collect, producing a cost model that suffers from incomplete information. As a result, even an optimal solution may be "wrong" in the sense that it is solving the wrong problem. In many operational settings, knowledgeable experts will already know, at least approximately, how a model should behave, and can express this knowledge in the form of low dimensional patterns: "high powered locomotives should pull intermodal trains" (because they need to move quickly) or "loaded C-141s should not be flown into Saudi Arabia" (for maintenance reasons). Unlike the literature on inverse optimization which uses observed actions to train the parameters of a cost model, we used exogenous patterns to guide the behavior of a model using a proximal point term that penalizes deviations from these patterns. Under the assumption that the patterns are derived from rational behaviors, we establish the conditions under which incorporating patterns will reduce actual costs rather than just the engineered costs. The effectiveness of the approach is demonstrated in a controlled, laboratory setting using data from a major railroad.

1. Introduction

Many operational problems that arise in practice offer sufficient complexity that it is difficult or impossible to completely model all the issues that are needed to produce realistic behaviors. The problem of optimizing the flows of locomotives for a major railroad provides a nice illustration. A railroad may prefer to use high-powered locomotives to pull service-sensitive intermodal trains, since these compete directly against trucks. If a lower-powered locomotive is used, the train arrives later and the resulting degradation in service can, over time, produce a loss of revenue. It is impossible to quantify the loss of revenue, and nor is it operationally possible to use only high-powered locomotives on these trains. It is possible to reflect this issue by adding an artificial cost when assigning a lower-powered locomotive to an intermodal train, but there are a dozen locomotive types and several dozen train types. This produces over 100 cost parameters which would have to be tuned to obtain acceptable performance.

It is very easy for a cost minimizing optimization model to produce solutions that are unacceptable to a knowledgeable expert, for reasons that are not reflected in the objective function. Other examples that we have encountered include:

- A truckload motor carrier likes to put certain types of drivers on longer loads because they have a high rate of quitting when they are put on shorter loads (the model is not capturing the economics of the driver).
- It is better to assign the printing jobs of a particular customer to the plant in Philadelphia because they can more easily send a representative to examine the samples (we are not modeling the cost of flying the representative to more distant plants).
- The military airlift command tries to avoid sending loaded C-141s (a large cargo aircraft) into certain regions because of the lack of maintenance facilities. If the plane requires maintenance, special equipment and expertise has to be flown in, which will delay the delivery of the load of freight (we are not modeling these additional maintenance costs).
- A shipping company tries to use 40 foot containers for loads into the Chicago area, since there are customers at that location that prefer this type of equipment.

The goal is to create models which produce realistic decisions that manage resources effectively. It is not enough to produce an optimal solution; the solution also has to be acceptable to people knowledgeable about the operations. The challenge is to develop a method that solves the dual challenge of minimizing costs while creating solutions that "look good" to experts. We assume that our expert is able to express desirable behaviors, but only in a relatively aggregate manner. For example, our expert might state "it is undesirable to use powerful locomotives to pull trains over

0740-817X © 2006 "IIE"

a particular section of track." Similar patterns (using the same problem context) might be of the form "we like to use a particular type of locomotive on intermodal trains" (reflects the power and efficiency of these locomotives and the service requirements of this type of train), "we like to take locomotives that came into a yard on train A and put it on outbound train B" (reflects missing information about the local yard configuration), and "we like to put locomotives at location A that are due for maintenance on trains heading east" (expresses the desire to move locomotives needing maintenance toward the shop location).

Traditionally, two strategies have been used to model these complex operational problems. The first is to simply use rule-based simulations where desired behaviors can be coded directly into the functions that control the system. This strategy is used in a vast array of simulation models. A limitation of this approach in applications such as those that arise in transportation and logistics is that the dimensionality of the decision vectors can be quite high, making it difficult to create manageable sets of rules that produce good solutions. Another limitation is that desired behaviors cannot always be written as hard rules; instead, it may be necessary to handle instructions such as "we *prefer* to put certain types of drivers on long loads" or "we *try* to avoid sending C-141s into certain regions."

The second strategy is to use a cost model with classical math programming algorithms. This strategy depends on our ability to formulate a good cost function. In the types of very complex applications that we have encountered, it is possible to develop good cost models, but they tend to suffer from missing information which simply does not capture (or properly quantify) issues that can be important in operations. Instead, modelers will resort to procedures which involve running the model, computing various performance statistics, and then putting costs (or bonuses) on behaviors that they want to decrease (or increase). The process of iteratively evaluating behaviors and tuning these artificial costs is ad hoc and time consuming. For example, penalizing one behavior in response to input from a user can produce an increase in another undesirable pattern of behavior. In actual applications, there can be several classes of patterns, and within a class there can be hundreds or thousands of instances of a pattern.

The problem of matching exogenously specified patterns is similar to the problem faced in statistics of choosing a set of parameters so that a model matches a given set of data. In our setting, we are trying to make an optimization model match a set of data. The problem is comparable to that addressed under the heading of inverse optimization (see Burton *et al.* (1997) for applications to shortest path problems and Zhang and Liu (1999), Sokkalingam *et al.* (1999), and Ahuja and Orlin (2001) for applications in the context of general linear programming and network problems). Inverse optimization takes a known cost function parameterized by unknown parameters and uses data to fit these parameters. In our setting, we have an approximate cost function where we may have inaccurate parameters, or the structure of the cost function itself may not be correct. Instead, we assume that we have exogenous data in the form of desired patterns of behavior. In complex problems, these patterns are almost always specified at a more aggregate level. Our use of simple patterns reflects a natural tendency of humans to express what they know about a system at different levels of aggregation (see Parsons (1996)).

Our strategy works as follows. In optimization, we would normally minimize a cost function which we can represent as $C(x)$. Now let ρ be a pattern vector, where an element of ρ would be an estimate of the flow (at some level of aggregation) that an exogenous user would expect from the model. These exogenous flows are targets, and may not themselves even represent feasible flows. If x represents the flows from the model, let $G(x)$ be the flows at the same level of aggregation as the pattern. Our expert would like to minimize the difference between the model flows, $G(x)$ and the flows that he expects based on his expert knowledge. We propose to combine these goals and minimize $C(x) + \theta\|G(x) - \rho\|$, where θ is a scaling term. The term $\|G(x) - \rho\|$ is typically the Euclidean norm which measures the difference between the model solution and the exogenous patterm. Increasing θ increases the emphasis on matching the pattern. In contrast with inverse optimization, we are not trying to achieve the right behavior by coming up with the right costs; instead, we are using our knowledge of appropriate behaviors to push the model toward these behaviors, but in a way that exploits our incomplete knowledge of the physics of the problem.

Objective functions such as these have long been studied under the umbrella of proximal point algorithms. Rockafellar (1976) first introduced the idea as a way of stabilizing algorithms for nondifferentiable functions. Ruszczyński (1987) introduced the idea of using proximal terms to stabilize stochastic gradient algorithms, where the cost function $C(x)$ at a given iteration would depend on a Monte Carlo sample. In both of these strategies, the proximal term penalizes the difference between the solution at a particular iteration n, and prior solutions. In our application, we penalize the difference between our solution and the exogenous pattern.

The paper makes the following contributions. 1) We introduce, apparently for the first time, the idea of using a proximal point framework to produce a model that better matches exogenously specified behaviors. Our strategy represents an alternative to inverse optimization, which seeks to find the cost function which, when minimized, produces a solution that is closer to what is observed. 2) We show that if the exogenous behaviors are based on a rational model (for example, they are the optimal solution based on a correct cost function), that under certain conditions the use of exogenous patterns will produce a better solution to the true (unobserved) cost function. 3) Using data from a major railroad, with a contrived "true" cost model and an

Downloaded by [Princeton University] at 07:52 04 January 2013

Downloaded by [Princeton University] at 07:52 04 January 2013

imperfect engineering cost model, we show that patterns can reduce true costs when the patterns are based on rational behavior. The resulting strategy is a practical method for improving model acceptance with considerably less effort than may be required by tuning the engineering cost model.

The paper is organized as follows. Section 2 describes a model for incomplete information and a methodology for improving model behavior using exogenously specified patterns. An important concept is the structure of the missing information and the structure of the exogenous patterns. Next, Section 3 establishes several theoretical properties of the modeling approach. Section 4 tests the concept experimentally using the data of a major railroad. Technical proofs are provided in the appendix.

2. Modeling incomplete information using patterns

We present the core modeling and algorithmic framework, using a specific locomotive application to illustrate the concepts, in Section 2.1. The key idea here is that there are unmeasurable and unobservable costs that produce undesirable behaviors in the model. We overcome this limitation using exogenously specified patterns of behavior. This modeling framework introduces two key issues: the structure of the patterns relative to the model (presented in Section 2.2), and the structure of the information relative to the patterns (Section 2.3).

2.1. *The basic model and solution strategy*

We present our model using a specific industrial application to illustrate the concepts. Assume we are optimizing the flows of locomotives for a major railroad (one of several projects where we have applied these ideas). For the purposes of our presentation, a locomotive is characterized by the following attribute vector:

$$a = \{\text{Locomotive type, Ownership, Location}\}$$

Here, “ownership” captures the property that a locomotive owned by one railroad may move trains onto the network of another railroad which may then use it for a period of time (while paying the owning railroad a fee). We act on the locomotive by assigning it to move a train. The attributes of the decision are the attributes of the train to which a locomotive is assigned, which might be:

$$d = \{\text{Train-ID, Origin, Destination, Tonnage, Train Type}\}$$

We let $\mathcal{A}$ be the set of potential attribute vectors, and $\mathcal{D}$ be the set of potential decision vectors, where $\mathcal{A}$ and $\mathcal{D}$ capture the level of detail that we are using inside our computer model. Our focus here is on decisions that act on a resource (moving a locomotive, repairing an aircraft), as opposed to decisions such as pricing which do not act on a resource. The explicit representation of attributes and decisions as vectors is central to the paper, since our ability to measure information often cannot be done at the same level of detail as we require in a model. We represent the *resource state vector* using:

$$\begin{aligned} R_a &= \text{The number of resources with attribute vector } a.\\ R &= \text{The resource state vector}\\ &= (R_a)_{a\in\mathcal{A}}.\end{aligned}$$

The decision vector is represented using:

$$\begin{aligned} x_{ad} &= \text{The number of times that we act on a}\\ &\quad\ \text{resource with attribute } a \text{ using decision } d.\\ x &= (x_{ad})_{a\in\mathcal{A},d\in\mathcal{D}}\end{aligned}$$

Decisions are constrained by standard conservation of flow

$$R_a(x) = \sum_{d\in\mathcal{D}} x_{ad} \tag{1}$$

as well as nonnegativity. The costs are given by

$$\begin{aligned} c_{ad} &= \text{The unit cost of acting on a resource with}\\ &\quad\ \text{attribute } a \text{ using decision type } d.\\ C(x) &= \text{The engineering cost function}\\ &= \sum_{a\in\mathcal{A}}\sum_{d\in\mathcal{D}} c_{ad}x_{ad}\end{aligned}$$

$C(x)$ is the standard engineering cost model consisting of components that we have estimated. Typically, there are two differences between the engineered costs and actual costs. The first is randomness that can be observed after the fact. For example, the cost of actually moving the locomotive depends on weather and delays. The actual fuel usage and time required (which impacts the pay of the crew) is random at the time the decision is made, but measurable after the fact. It is common practice to approximate these random variables using estimates of the mean.

The second difference is unobservable costs. For example, if we use a less-powerful locomotive, the train will move slower, reducing the quality of service. This will contribute to a perception of poor service with a possible downstream impact on revenue. This effect is virtually impossible to measure, so railroads will institute a policy of using high-powered locomotives on certain types of service-sensitive trains.

We can represent these different costs by writing

$$\tilde{C}(x) = C^o(x, c^k, \hat{c}) + C^u(x, c^u) \tag{2}$$

where

$$\begin{aligned} \tilde{C}(x) &= \text{The actual costs resulting from a vector of}\\ &\quad\ \text{decisions.}\\ C^o(x, c^k, \hat{c}) &= \text{The observable costs, which depend on the}\\ &\quad\ \text{decision vector } x\text{, a set of known (deter-}\\ &\quad\ \text{ministic) cost parameters } c^k\text{, and a set}\\ &\quad\ \text{of random (but observable after the fact)}\\ &\quad\ \text{costs } \hat{c}.\\ C^u(x, c^u) &= \text{The unobservable (and unmeasurable) costs.}\end{aligned}$$

Downloaded by [Princeton University] at 07:52 04 January 2013

Our engineering cost function is given by

$$C(x) = C^o(x, c^k, \bar{c}) \tag{3}$$

where $\bar{c}$ is a point estimate of the unknown random vector $\hat{c}$ used in Equation (2). We use the compact notation $C^o(x)$ for $C^o(x, c^k, \hat{c})$. Keep in mind that $C^o(x)$ is random, while $C(x)$ is deterministic.

The focus of this paper is on the unmeasurable costs $C^u(x, c^u)$. Since it is unmeasurable, we seek to obtain more realistic behaviors by guiding the model using exogenous patterns of behavior. For example, we might specify:

ρ_{ad} = The fraction of time that a resource with attribute a should be acted on with a decision of type d.

$\rho = (\rho_{ad})_{a \in \mathcal{A}, d \in \mathcal{D}}$

We refer to ρ as the *pattern vector*. If x_{ad} is the flow inside the model, then we define

$$x^p_{ad} = \text{The pattern flow} \tag{4}$$

$$= R_a \rho_{ad} \tag{5}$$

The pattern flow can be viewed as what we should be doing in the model if we precisely matched the exogenous pattern. In practice this may be infeasible (as a result of flow conservation, we cannot always match a pattern) as well as undesirable. The pattern ρ can be viewed as a guide, to be balanced against the engineering costs. We measure the difference between model flows and pattern flows using

$$H(x, \rho) = \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} (x_{ad} - R_a \rho_{ad})^2 \tag{6}$$

It is useful to think of $H(x, \rho)$ as the "happiness function" since minimizing it usually makes people in operations more satisfied with the solution. We combine this function with our traditional cost function to produce a modified optimization problem

$$x^*(\theta) = \arg \min_{x \in \mathcal{X}} C(x) + \theta H(x, \rho) \tag{7}$$

where $\mathcal{X}$ is the feasible region and is assumed to be compact. θ is a positive scaling factor that weights the pattern metric with respect to the cost function.

Equation (7) is equivalent to the proximal point framework first proposed in Rockafellar (1976) to help stabilize the convergence of algorithms for nondifferentiable problems. Ruszczyński (1987) also showed how this framework could be used to stabilize the convergence of stochastic gradient algorithms for stochastic optimization problems. In both settings, the proximal term is contributing information that is not captured by the cost function used at each iteration. Equation (7) can also be viewed as a type of Lagrangian relaxation with Lagrange multiplier θ, but this interpretation is typically used when relaxing a constraint. In our setting, we view matching the pattern ρ as more of a goal, and feel that the proximal point framework is the correct way to view the problem.

The ultimate goal of the strategy is to find x that minimizes $\tilde{C}(x)$. Since we cannot compute $\tilde{C}(x)$, we instead compute $x^*(\theta)$ for a value of θ that appears (to a knowledgeable user) to strike a balance between minimizing the computed costs and producing a solution that "appears reasonable." We test this idea in Section 4 by first creating an assumed function $\tilde{C}$ and then creating an engineering cost function $C(x)$ that differs from $\tilde{C}$ by a random quantity (we consider two cases: zero mean differences and biased differences). This controlled experiment is supplemented by a theoretical analysis of the behavior of the model, given in Section 3. Before proceeding to these sections, we have to address the issue of aggregation and information representation. Section 2.2 shows how to handle patterns at different levels of aggregation. Section 2.3 then discusses the issues that arise with the structure of the missing information. The experimental work in Section 4 looks at the effectiveness of the algorithmic strategy for different pattern and information structures.

2.2. *Model and pattern aggregation*

It is typically the case that a knowledgeable expert will express his/her expertise by criticizing a specific behavior of the model. Continuing our locomotive example, a user might note that we should use more high- powered (or less lower-powered) locomotives on a specific type of service sensitive trains. Other patterns we have observed in a railroad context sound like "locomotives that arrive on a particular inbound train should be assigned to a particular outbound train" or "locomotives needing maintanence (at a particular location) should be assigned to trains moving east." These patterns are expressed at some level of aggregation. In general, there may be multiple pattern classes. We represent these using

$\mathcal{P}$ = Set of patterns.

G^p_a = Attribute aggregation function using pattern $p \in \mathcal{P}$ that acts on the attribute vector a.

$\mathcal{A}^p$ = Aggregated attribute space, where $G^p_a : \mathcal{A} \to \mathcal{A}^p$.

G^p_d = Attribute aggregation function using pattern p that acts on the decision vector d.

$\mathcal{D}^p$ = Aggregated decision space, where $G^p_d : \mathcal{D} \to \mathcal{D}^p$.

For notational compactness, we use $\hat{a}$ and $\hat{d}$ to represent aggregated attributes and decisions when the specific level of aggregation (alternatively, the specific pattern) is not important. If the attribute vector and decision is:

a = {Location, Time, Locomotive type, Horsepower, Length, Inbound train}

d = {Train ID, Train type, Tonnage, Destination, Delay}

then one pattern might be

$G^1_a(a)$ = {Location, -, Locomotive type, -, -, -}

$G^1_d(d)$ = {-, Train type, -, -, -}

Downloaded by [Princeton University] at 07:52 04 January 2013

Another pattern might be

$$G_a^2(a) = \{\text{Location, -, -, -, -, Inbound train}\}$$
$$G_d^2(d) = \{\text{-, -, -, Destination, -}\}$$

For a given pattern, the aggregated flows are represented using

$$G^p(x^p) = \left\{x^p_{\hat{a}\hat{d}}\right\}_{\hat{a}\in\mathcal{A}^p, \hat{d}\in\mathcal{D}^p_{\hat{a}}}$$

where for each $\hat{a} \in \mathcal{A}^p$ and $\hat{d} \in \mathcal{D}^p_{\hat{a}}$ we define the *pattern flow*:

$$x^p_{\hat{a}\hat{d}} = \sum_{a\in\mathcal{A}} \sum_{d\in\mathcal{D}_a} x^p_{ad} I_{\{G^p_a(a)=\hat{a}\}} I_{\{G^p_d(d)=\hat{d}\}}$$

We also write

$$R^p_{\hat{a}}(x) = \sum_{a\in\mathcal{A}} R_a I_{\{G^p_a(a)=\hat{a}\}}$$

to be the total flow through aggregated attribute $\hat{a}$.

When the exogenous patterns are measured at some level of aggregation, we have to modify our definition of the distance metric $H(x, \rho^p)$ to work at the same level of aggregation as the pattern. For this reason we define

$$H^p(x, \rho^p) = \sum_{\hat{a}\in\mathcal{A}^p} \sum_{\hat{d}\in\mathcal{D}^p} \left(x^p_{\hat{a}\hat{d}} - R^p_{\hat{a}}(x)\rho^p_{\hat{a}\hat{d}}\right)^2$$

To handle multiple pattern classes, our decision function looks like

$$x^*(\theta) = \arg\min_{x\in\mathcal{X}} C(x) + \sum_{p\in\mathcal{P}} \theta^p H^p(x, \rho^p) \tag{8}$$

We assume that the optimization problem in Equation (8) can be solved using an existing algorithm. We note that $H^p(x, \rho^p)$ is separable in the aggregated flows. It may be helpful to replace $H^p(x, \rho^p)$ (which is nonlinear) with a piecewise linear penalty function.

Section 3 investigates the theoretical properties of this decision function. Before moving to this stage, the next section looks at the structure of the missing information.

2.3. *Pattern aggregation and the structure of missing information*

The effectiveness of using a particular pattern ρ^p to help minimize the unobservable $\tilde{C}(x)$ depends on the nature of the missing information. Assume that $\tilde{C}(x)$ is given by

$$\tilde{C}(x) = \sum_{a\in\mathcal{A}} \sum_{d\in\mathcal{D}} \tilde{c}_{ad} x_{ad}$$

where $\tilde{c}_{ad}$ are unobservable costs per unit of flow. We further assume that

$$\tilde{c}_{ad} = c_{ad} + c^u_{ad}$$

We have to recognize that the unobservable costs c^u_{ad} may depend on a and d at some level of aggregation, which may or may not be the same as the level of aggregation of the pattern that is trying to reflect these unobservable costs. For example, we may prefer that high-powered locomotives be used to pull service-sensitive intermodal trains, but in reality it is only intermodal trains going from Chicago to the west coast that really matter. Here, the missing information is more detailed than the pattern. Alternatively, we wish to put locomotives that arrive on one inbound train on a particular outbound train (reflecting costs at the terminal) when in fact all we really want to do is to ensure that locomotives on trains arriving from a particular terminal are put on trains that have a particular destination. Here, the missing information is at a more aggregate level than the pattern.

The relationship between the level of detail captured by the exogenous pattern, and the level of detail in the missing information, determines our ability to optimize $\tilde{C}$ using an exogenous pattern ρ^p. Given the importance of this concept, we define

$I^u(c^u) =$ The highest level of aggregation at which the missing information is completely captured.

To keep the notation relatively compact, we let $\mathcal{P}$ represent all possible levels of aggregation of patterns and information, and we assume that the patterns are ordered, as in $\mathcal{P} = \{1, 2, \ldots, P\}$ where $p = 1$ is the most disaggregate and $p = P$ is the most aggregate. $I^u(c^u) = 2$, then, means that the missing information is at the same level of aggregation as pattern class $p = 2$. $I^u(c^u) > p$ means that our patterns are more detailed than our missing information.

The interaction between the structure of patterns and missing information is investigated in the experimental work in section 4.

3. Model behavior

The goal of this section is to study some of the theoretical properties of our model. Recall that in Equation (2) we split the real-world objective $\tilde{C}$ into its observable and unobservable components C^o and C^u. Our expert knowledge, captured in the form of patterns, is assumed to reflect the missing information in C^u, but will not in general be optimal. To analyze some of the properties of our formulation, we are going to assume that our "experts" are, in fact, perfect and provide optimal solutions to $\tilde{C}$. We then assume that some of this information is lost through aggregation and then transferred to the model in the form of patterns. In this section, we study the properties of a model that incorporates patterns from a "perfect expert." Section 3.1 provides the setting for our study, and Section 3.2 summarizes a series of properties.

3.1. *Model setting*

We begin by assuming that we can optimally solve:

$$x^{p*} = \arg\min_{x^p\in\mathcal{X}} \tilde{C}(x^p) \tag{9}$$

Downloaded by [Princeton University] at 07:52 04 January 2013

In reality humans do not solve this model optimally. x^{p*} represents the decisions in our laboratory construction of a pattern database characterizing expert knowledge. We denote the real-world objective at the optimal solution x^{p*} by:

$$\tilde{C}^* = \tilde{C}(x^{p*}) \tag{10}$$

We now assume that our perfect solution is lost through aggregation when it is represented as a pattern. The corresponding optimization model with the pattern metric is given by:

$$x^*(\theta) = \arg\min_{x\in\mathcal{X}} C(x) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}} - x^{p*}_{\hat{a}} \right\|^2 \tag{11}$$

where for succinctness we use the following notation for the pattern metric:

$$\left\| x_{\hat{a}} - x^{p*}_{\hat{a}} \right\|^2 = \sum_{\hat{d}\in\hat{\mathcal{D}}_{\hat{a}}} \left\| x_{\hat{a}\hat{d}} - x^{p*}_{\hat{a}\hat{d}} \right\|^2 \quad \forall \hat{a} \in \hat{\mathcal{A}}$$

Note that we have fixed the level of aggregation n in the model given in Equation (11) and $\hat{a} = G^p_a(a)$, $a \in \mathcal{A}$, $\hat{\mathcal{A}} = \mathcal{A}^p$ and $\hat{\mathcal{D}}_{\hat{a}} = \mathcal{D}^p_{\hat{a}}$. Solving the model in Equation (11) using the traditional approach (that is $\theta = 0$) yields a suboptimal solution to the model $\tilde{C}$ since the objective $\tilde{C}$ consists of cost parameters that are not considered in C. Our goal in using the methodology given in Equation (11) is to get a solution that when evaluated by $\tilde{C}$ is closer to $\tilde{C}^*$ given in Equation (10) than when using the solution obtained from just the traditional method. Since Equation (11) only has access to the aggregated patterns, the use of patterns seems intuitively useful but the actual benefits are not obvious.

3.2. *Model properties*

We make the following assumptions:

A.1 The cost function C is convex in x.

A.2 The level of aggregation at which the pattern metric is represented is fixed at p.

A.3 If the minimizer of C over the compact set $\mathcal{X}$ is not unique, that is the set $\mathcal{X}(0) = \arg\min_{x\in\mathcal{X}} C(x)$ consists of more than one element, then we set the optimal solution to $\min_{x\in\mathcal{X}} C(x)$ as:

$$x^*(0) = \arg\min_{x\in\mathcal{X}(0)} \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}} - x^{p}_{\hat{a}} \right\|^2$$

A.4 As our goal is to capture information contained in C^u through the pattern flows x^{p*} we assume:

$$C^u(x^{p*}) < C^u(x) \quad \forall x \in \mathcal{X}(0)$$

A.5 All the elements of the vector $\{|x_{\hat{a}}(0) - x^{p*}_{\hat{a}}|\}_{\forall \hat{a}\in\hat{\mathcal{A}}}$, for each $x(0) \in \mathcal{X}(0)$, are nonzero.

A.6 The feasible region $\mathcal{X}$ is compact (closed and bounded).

Assumption A.5, which says that the flow obtained from minimizing costs is never exactly the same as a pattern, is a purely technical assumption needed to prove uniqueness and that as θ increases, we move monotonically toward the pattern flow. A.5 can always be enforced by simply dropping from the pattern database any pattern which exactly matches the cost-minimizing flow.

We adopt the following convention for a vector of decision variables. Two vectors are equal if and only if all of its scalar components are identical. We say a vector is monotonically increasing (decreasing) in θ if all its scalar elements are monotonically increasing (decreasing) in θ.

The following theorem establishes that $x^*(\theta)$ gets closer to x^{p*} as θ increases, and that this produces a correspondingly higher objective function $C(x^*(\theta))$. The theorem claims that $\|x^*(\theta) - x^{p*}\|$ and $C(x^*(\theta))$ are strictly monotone in θ.

Theorem 3.1. *We have for* $0 < \theta_1 < \theta_2 < \infty$*:*

$$\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 < \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2, \tag{12}$$

and:

$$C(x^*(\theta_1)) < C(x^*(\theta_2)). \tag{13}$$

The proof of this theorem is quite technical, and is given in the appendix. For our purposes, it is sufficient to demonstrate simple monotonicity:

Proposition 3.1. *For any two scaling factors θ_1 and θ_2 such that* $0 < \theta_1 < \theta_2 < \infty$ *we have:*

$$C(x^*(\theta_1)) \leq C(x^*(\theta_2)). \tag{14}$$

and

$$\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 \leq \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2. \tag{15}$$

Proof. We prove Equation (14) by contradiction. Assume that $C(x^*(\theta_1)) > C(x^*(\theta_2))$. $x^*(\theta_1)$ is optimal to the model whose mathematical objective is $C(x^*(\theta_1)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\|^2$. Since $x^*(\theta_2)$ is a feasible solution to the same problem we have:

$$\begin{aligned} &C(x^*(\theta_1)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2 \\ &\quad \leq C(x^*(\theta_2)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 \end{aligned}$$

which implies:

$$\begin{aligned} &C(x^*(\theta_1)) - C(x^*(\theta_2)) \\ &\quad \leq \theta_1 \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2 \right). \end{aligned} \tag{16}$$

Since $C(x^*(\theta_1)) - C(x^*(\theta_2)) > 0$ by assumption the right hand side of Equation (16) is strictly greater than 0. We

Downloaded by [Princeton University] at 07:52 04 January 2013

apply the same analysis for the model with scaling factor θ_2 and we see that

$$C(x^*(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 \le C(x^*(\theta_1)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2.$$

Rearranging the terms we get:

$$C(x^*(\theta_1)) - C(x^*(\theta_2)) \ge \theta_2 \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2 \right). \quad (17)$$

Combining Equations (16) and (17) we have:

$$\theta_2 \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2 \right) \le \theta_1 \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2 \right),$$

that is:

$$(\theta_2 - \theta_1) \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2 \right) \le 0. \quad (18)$$

From Equation (16) we see that $(\sum_{\hat{a}\in\hat{\mathcal{A}}} \|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\|^2) > 0$. Also we have $(\theta_2 - \theta_1) > 0$ thus violating Equation (18). Thus we have the $C(x^*(\theta_1)) \le C(x^*(\theta_2))$ as stated in Equation (14). We next prove Equation (15) by contradiction. Assume that we have

$$\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 > \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2.$$

Using Equation (14) it follows that:

$$C(x^*(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 > C(x^*(\theta_1)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2. \quad (19)$$

Now $x^*(\theta_2)$ is optimal to the model with the scaling factor equal to θ_2 and since $x^*(\theta_1)$ is a feasible solution we have:

$$C(x^*(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 \le C(x^*(\theta_1)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2.$$

This violates the relation in Equation (19). Thus we have:

$$\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}}\right\|^2 \le \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}}\right\|^2.$$

as stated in Equation (15). ■

The next proposition proves the asymptotic convergence in the scaling factor θ of the model flows to the pattern flows.

Proposition 3.2. *As the scaling factor $\theta \to \infty$ we have*

$$\{x^*_{\hat{a}}(\theta)\}_{\forall \hat{a}\in\hat{\mathcal{A}}} \to \left\{x^{p*}_{\hat{a}}\right\}_{\forall \hat{a}\in\hat{\mathcal{A}}}.$$

Proof. Since x^{p*} is a feasible solution to the model with a pattern metric we have:

$$C(x^*(\theta)) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 \le C(x^{p*}) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^p_{\hat{a}} - x^p_{\hat{a}}\right\|^2.$$

Since the feasible region $\mathcal{X}$ is compact by assumption A.6 it follows that:

$$C(x^*(\theta)) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 \le C(x^{p*}) < \infty.$$

As $\theta \to \infty$, we must have $\sum_{\hat{a}\in\hat{\mathcal{A}}} \|x^*_{\hat{a}}(\theta) - x^p_{\hat{a}}\|^2 \to 0$ which implies $\{x^*_{\hat{a}}(\theta)\}_{\forall \hat{a}\in\hat{\mathcal{A}}} \to \{x^{p*}_{\hat{a}}\}_{\forall \hat{a}\in\hat{\mathcal{A}}}$.

The following proposition establishes that if engineering costs are captured perfectly, and we represent patterns at the same structural level as the incomplete information, the real cost function $\tilde{C}(x^*(\theta))$ evaluated using $x^*(\theta)$ (given by Equation (11), which only uses the aggregated pattern) approaches the real optimal solution $\tilde{C}^*$. ■

Proposition 3.3. *If we have perfect engineering costs, that is, $C(x) = C^o(x)$ (there is no randomness in the estimated cost elements) and the information structure index $I^u(c^u)$ equals the level of aggregation of the pattern metric, that is $I^u(c^u) = p$, then we have:*

$$\lim_{\theta\to\infty} \tilde{C}(x^*(\theta)) = \tilde{C}^*. \quad (20)$$

Proof. We note that proposition 3.3 implies that we are able to capture all the incomplete information by solving the model with a pattern metric if we let the scaling factor increase indefinitely. x^{p*} is the optimal solution to the problem:

$$\min_{x\in\mathcal{X}} C^o(x) + C^u(G(x)).$$

where G is the aggregation function. Since C^u is a function of only the aggregated decision variables at level p and because $I^u(c^u) = p$, we can express the real-world optimization problem in the form below:

$$x^{p*} = \arg \min_{x\in\mathcal{X}\cap\{\hat{x}=\hat{x}^{p*}\}} C^o(x). \quad (21)$$

Now consider the optimization model with a pattern metric:

$$x^*(\theta) = \arg\min_{x\in\mathcal{X}} C(x) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x^*_{\hat{a}}(\theta) - x^{p*}_{\hat{a}}\right\|^2. \quad (22)$$

Downloaded by [Princeton University] at 07:52 04 January 2013

Since the pattern metric is a function of only the aggregated decision variables we can rewrite the model in Equation (22) as:

$$x^*(\theta) = \arg \min_{x \in \mathcal{X} \cap \{\hat{x}(\theta) = \hat{x}^*(\theta)\}} C(x) \tag{23}$$

From proposition 3.2 we have:

$$\lim_{\theta \to \infty} \{x^*_{\hat{a}}(\theta)\}_{\forall \hat{a} \in \hat{\mathcal{A}}} = \left\{x^{p*}_{\hat{a}}\right\}_{\forall \hat{a} \in \hat{\mathcal{A}}}. \tag{24}$$

Incorporating Equation (24) into Equation (23) we see that:

$$\lim_{\theta \to \infty} x^*(\theta) = \arg \min_{x \in \mathcal{X} \cap \{\hat{x}(\theta) = \hat{x}^{p*}\}} C(x). \tag{25}$$

From our assumption that $C(x) = C^o(x)$ we see that the models indicated by Equations (21) and (25) are exactly identical and hence we must have:

$$\lim_{\theta \to \infty} x^*(\theta) = x^{p*}.$$

Thus it follows that:

$$\lim_{\theta \to \infty} \tilde{C}(x^*(\theta)) = \tilde{C}(x^{p*}) = \tilde{C}^*.$$

■

This section has established that if patterns are rational (that is, they minimize the real function) and the patterns are captured at the same level of aggregation as the incomplete information, then incorporating patterns will produce optimal solutions in the real world. In practice, patterns are almost always aggregated, producing a loss of information. In this case, we can not even say that $C(x^*(\theta))$ improves monotonically with θ. For these more realistic settings, we turn to experimental research.

4. Experimental design and results

We present here an experiment with real-world data representing the locomotive operations of a large freight railroad. Although we use this single application to illustrate our ideas, it is important to emphasize that this represents a major industrial application. The data, attributes and patterns are real.

Consistent with the approach taken in Section 3 we derive our "patterns" by first solving, to optimality, a linear program which is assumed to include all information:

$$x^* = \arg \min_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} c^o_{ad} x_{ad} + \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}} c^u_{ad} x_{ad}$$

where $\{c^o_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ and $\{c^u_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ are vectors of cost parameters characterizing C^o and C^u. Recall that $\{c^o_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ represents a realization of a random measurement of costs.

Since the function C^u is a component of $\tilde{C}$ we assume that in general C^u is a function of only the decision variables corresponding to states belonging to a set $\tilde{\mathcal{A}}$, where $\tilde{\mathcal{A}}$ is a subset of the space of resource attributes $\mathcal{A}$. Thus C^u in general is a function of the vector of decision variables $\{x^*_{ad}\}_{a \in \tilde{\mathcal{A}}, d \in \mathcal{D}_a}$. We aggregate $\{x^*_{ad}\}_{a \in \tilde{\mathcal{A}}, d \in \mathcal{D}_a}$ using an aggregation function $G^p = (G^p_a, G^p_d)$ to generate the vector of pattern flows x^{p*} for pattern p.

We then solve the following engineering cost model with the pattern metric:

$$x^*(\theta) = \arg \min_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}_a} c_{ad} x_{ad} + \theta \sum_{\hat{a} \in \hat{\mathcal{A}}} \sum_{\hat{d} \in \hat{\mathcal{D}}_{\hat{a}}} (x_{\hat{a}\hat{d}} - x^{p*}_{\hat{a}\hat{d}})^2 \tag{26}$$

where $\{c_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ is the vector of cost parameters characterizing the linear cost function C. In our experiment the vector of costs $\{c^o_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ is derived by randomly perturbing $\{c_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ by a vector $\{\hat{c}_{ad}\}_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ which may be biased meaning that the mean of the random perturbation may not necessarily be 0. We note that the optimization problem in Equation (26) is a surrogate for $\tilde{C}$.

We denote $x^* = (x^*_{ad}(0))_{a \in \mathcal{A}, d \in \mathcal{D}_a}$ as the optimal solution to solving the optimization model without the pattern metric (that is, $\theta = 0$ in Equation (26)). We claim that by solving the optimization model with a pattern metric as indicated in Equation (26) we are able to get a positive improvement ratio defined by:

$$\eta^{IMP}(\theta) = \frac{\tilde{C}(x^*(0)) - \tilde{C}(x^*(\theta))}{\tilde{C}(x^*(0)) - \tilde{C}^*}$$

An improvement ratio close to 1 means that for a particular value of θ the optimality gap with respect to the real-world objective $\tilde{C}$ is close to 0. In the following sections we present the experimental setup detailing the construction of the cost functions C, $\tilde{C}$ and C^u, the attribute space $\tilde{\mathcal{A}}$, the aggregation functions and the results of our research.

4.1. *Experimental setup*

Our experimental setting is a locomotive scheduling operation for a large railroad. The resource state a for the locomotive consists of the following attributes:

$$a = \{\text{Locomotive Type, Ownership, Location}\}$$

and the decision d pertaining to the state a is as shown:

$$d = \{\text{Train-ID, Origin, Destination, Tonnage, Train Type}\}$$

The engineering costs are reflected in C^o (or C).

A challenge faced by railroads is managing locomotives that are owned by other railroads, known as "foreign power." The rules governing how railroad A manages the locomotives owned by railroad B can be very complex, dependent on measurable quantities (the location and ownership of the locomotive) and incomplete information (strategies for moving foreign power back to its home railroad, rules that govern the cost of using locomotives owned by other railroads). We propose to use a pattern metric to guide the management of foreign power. The "Ownership" attribute in a is an indication of whether the locomotive is owned

Downloaded by [Princeton University] at 07:52 04 January 2013

by the company or by a foreign railroad. Thus the attribute space $\tilde{\mathcal{A}}$ for our experiment consists of all the resource attribute vectors in $\mathcal{A}$ whose "Ownership" attribute indicates that the locomotive is owned by a foreign railroad. We use $\hat{\mathcal{A}}$ to denote the space of the aggregated states $\hat{a}$ derived from the set $\tilde{\mathcal{A}}$ and $\hat{\mathcal{D}}_{\hat{a}}$ to denote the space of the aggregated decisions $\hat{d}$ derived for each aggregated state $\hat{a} \in \hat{\mathcal{A}}$.

The experimental data were obtained from a major railroad and represent the data for a week. The data consisted of 10685 train segments (each segment is a unique "Train-ID") connecting 540 locations in the network, 17 train types, 10 locomotive types and 6 types of ownership (including the company-owned type). Of the 10 locomotive types, 8 of them pertained only to company-owned locomotives whereas the remainder pertained to the locomotives belonging to foreign railroads.

Generating C^o and C: The modeler's cost function C is calibrated based on input from the people in locomotive operations for the railroad whose data we use in our experiment. Since the calibration is only approximate we assume that C is different from C^o. We consider two types of observable costs, a positive cost (the cost of assigning locomotives to trains) and a negative cost (the reward of assigning power to a train). The cost vector c^o_{ad} is derived from c_{ad} by perturbing it with a random component $\hat{c}_{ad}$, $\forall a \in \mathcal{A}$, $d \in \mathcal{D}_a$. We consider two separate scenarios under which we generate the random components $\hat{c}_{ad}$, $\forall a \in \mathcal{A}$, $d \in \mathcal{D}_a$.

Scenario 1: Zero-mean Noise: The random perturbation is generated based on these simple rules:

1. The range of values for c^o_{ad} is set such that the positive cost parameters are allowed to deviate uniformly up to ±75% of their respective values in C and the negative cost parameters are allowed to deviate uniformly up to ±25% of their respective values in C.
2. All the elements of the unobservable costs $\hat{c}_{ad}$ are assumed to be uniformly distributed within the range specified above such that $c^o_{ad} = c_{ad} + \hat{c}_{ad}$ and $\mathbb{E}\hat{c}_{ad} = 0$.

Scenario 2: Biased Noise: In this case we assume that as a modeler we are biased in calibrating the engineering costs. In this case the range of values for c^o_{ad} is set such that the positive cost parameters are allowed to deviate uniformly up to +150% of their respective values in C and the negative costs are allowed to deviate uniformly up to −80% of their corresponding values in C.

Generating C^u (or c^u): The cost vector c^u_{ad}, $\forall a \in \tilde{\mathcal{A}}$, $d \in \mathcal{D}_a$, is generated based on an actual historical database of decisions regarding management of foreign locomotive power. In real-world operations we can observe foreign locomotive movements on a particular corridor of the rail network. The pattern flows representing these foreign locomotives are characterized by the state $\hat{a}$ as shown below:

$$\hat{a} = G_a(a) = \{\text{Ownership, Location}\}, \quad a \in \tilde{\mathcal{A}}, \tag{27}$$

and the decision $\hat{d}$ given by:

$$\hat{d} = G_d(d) = \{\text{Destination}\}, \quad d \in \mathcal{D}_a, \quad a \in \tilde{\mathcal{A}}. \tag{28}$$

Table 1. Patterns flows: experimental scenarios. Represents the information captured at the level of aggregation of the pattern flows with respect to the structural information underlying c^u, the vector of cost parameters characterizing C^u

Information structure	*Pattern aggregation (index of aggregation level)*		
($I^u(c^u)$)	*O-L-D-ID (1)*	*O-L-D (2)*	*O-L (3)*
O-L-D-ID (1)	Complete	Loss	Loss
O-L-D (2)	Redundant	Complete	Loss

A file of historical priors representing foreign locomotive movements $\rho^p_{\hat{a}\hat{d}}$ is constructed from actual historical data for the pattern $(\hat{a}, \hat{d})$ indicated in Equations (27) and (28). If $\rho^p_{\hat{a}\hat{d}} > 0$ then $c^u_{ad} = \beta * \rho^p_{\hat{a}\hat{d}}$ for all $a \in \tilde{\mathcal{A}}$ such that $G_a(a) = \hat{a}$ and for all $d \in \mathcal{D}_a$ such that $G_d(d) = \hat{d}$ (in other cases c^u_{ad} is assigned a positive value). $\beta < 0$ is a parameter that scales the patterns into cost units (the more frequently a pattern is used, the more negative the cost and hence the pattern is encouraged). In this case the cost parameters pertaining to C^u are a function of the attributes "Ownership" (O) and "Location" (L) of the locomotive resource and attribute "Destination" (D) of any train originating at the same location as the locomotive. In our experiment the O-L-D has a level of aggregation 2. Thus the structural function conveying the information underlying the cost parameters characterizing C^u is given by $I^u(c^u) = 2$.

We also consider generating a vector of costs c^u at the $p = 1$ level denoted by O-L-D-ID which consists of the attributes "Ownership" and "Location" of the locomotive resource and besides the attribute "Destination" of the train also includes the attribute "Train-ID" (ID). In this case we generate $c^u_{ad} = \beta * \beta_{ID} * \rho^p_{\hat{a}\hat{d}}$ for all $a \in \tilde{\mathcal{A}}$ such that $G_a(a) = \hat{a}$ and for all $d \in \mathcal{D}_a$ such that $G_d(d) = \hat{d}$. β_{ID} is a random multiplier uniformly generated in the interval [0.2, 2] and is a function of the unique string characterized

Table 2. Zero mean noise scenario: imperfect engineering costs. Each table entry consists of (improvement ratio, θ)

Improvement ratio and corresponding value of θ			
Information structure	*Pattern aggregation*		
	O-L-D-ID	*O-L-D*	*O-L*
O-L-D-ID	(0.754,10000)	(0.417,10000)	(0.108,10000)
O-L-D	—	(0.585,10000)	(0.403,10000)

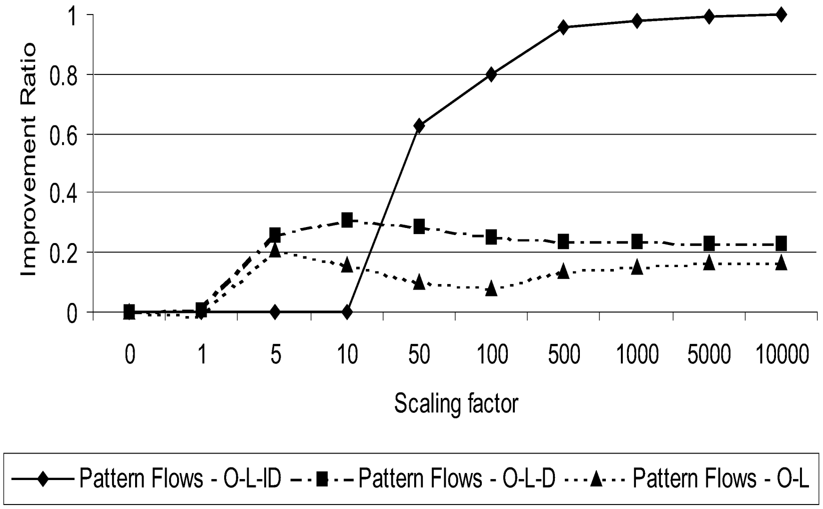

Fig. 1. Biased noise scenario: Improvement Ratio η^{IMP} with perfect engineering costs and foreign locomotive behavior generated at O-L-D-ID level.

Downloaded by [Princeton University] at 07:52 04 January 2013

by the attributes "Train-ID" and "Destination". In this case $I^u(c^u) = 1$.

When we generate our "pattern database" if we generate the costs pertaining to foreign locomotive movements at the O-L-D level the pattern flows of interest are represented only at the O-L-D (complete information) and O-L (loss of information) level. On the other hand if we generate the costs at the O-L-D-ID level then the pattern flows of interest are represented at the O-L-D-ID (complete information) and the O-L-D (loss of information) and O-L (loss of information) levels.

Thus in our methodology of capturing incomplete information conveyed by the unknown function C^u the pattern flows are represented at the same level or a higher level of aggregation than the level at which we generated the costs governing foreign locomotive movements. This approach is captured in Table 1.

4.2. *Summary of results*

All the models are linear programming problems and are solved on a SunOS5.8 platform using LOQO which utilizes interior-point methods to solve linear and quadratic programming problems (Vanderbei (1999)). LOQO uses an AMPL interface (Fourer *et al.*, 1993) to read in the model and data.

The effect of the scaling factor θ depends on the structure of the patterns and the missing information. Figure 1 shows that when the structure of patterns and missing information is the same, then the improvement ratio increases monotonically with θ. When the patterns are more aggregate than the missing information, then there is an optimal value of θ, and the improvement can actually be reduced if θ is set too large. When the pattern flows are represented at the aggregated level O-L-D the improvement ratio increases monotonically until $\theta = 10$. At the aggregated level O-L the best value of θ is around 5.

Given this behavior, we used two choices for θ when running our experiments. If the pattern aggregation was at the same level of detail as the information structure, we used $\theta = 10,000$ to force a high match to the pattern. We also used a high value of θ when the engineering cost parameters are estimated with no noise, or with zero-mean noise.

We present the results of our experiment in Tables 2–5. Each cell in these tables represents a combination of the

Table 3. Zero mean noise scenario: perfect engineering costs. Each table entry consists of (improvement ratio, θ)

Information structure	*Improvement ratio and corresponding value of θ* *Pattern aggregation* *O-L-D-ID*	*O-L-D*	*O-L*
O-L-D-ID	(0.997,10000)	(0.524,10000)	(0.155,10000)
O-L-D	—	(1.000,10000)	(0.745,10000)

Table 4. Biased mean noise scenario: perfect engineering costs. Each table entry consists of (improvement ratio, θ). Value of θ is optimized to produce the best improvement ratio

Information structure	*Improvement ratio and corresponding value of θ* *Pattern aggregation* *O-L-D-ID*	*O-L-D*	*O-L*
O-L-D-ID	(0.997,10000)	(0.313,10)	(0.210,5)
O-L-D	—	(1.000,10000)	(1.000,10000)

Downloaded by [Princeton University] at 07:52 04 January 2013

Table 5. Biased mean noise scenario: imperfect engineering costs. Each table entry consists of (improvement ratio, θ). Value of θ is optimized to produce the best improvement ratio

Improvement ratio and corresponding value of θ			
Information structure	*Pattern aggregation*		
	O-L-D-ID	*O-L-D*	*O-L*
O-L-D-ID	(0.240,10000)	(0.066,10)	(0.044,5)
O-L-D	—	(0.083,10000)	(0.082,10000)

structure of the incomplete information, and the level of aggregation in the pattern. Thus, if the information structure is "O-L-D-ID" then it means that the incomplete information depends on all four of these attributes. If the pattern aggregation is "O-L-D" or "O-L" then there is some loss of information. Each cell shows the improvement ratio (where values close to 1.0 mean that we are matching the optimal solution of the real objective function), and the value of θ used to produce the results.

The tables show, as we expected, that the improvement ratio is at or very near 1.0 when the pattern aggregation is at the same level as the information structure, with the exception of the extreme case where we used biased estimates with imperfect engineering costs (Table 5). Elsewhere we notice, as we would expect, a degradation in the improvement ratio as the patterns become more aggregated.

These experiments demonstrate, for this dataset, that pattern matching will provide the greatest value if the pattern database captures the structure of the missing information. Often, the structure of the pattern database is restricted by the information that has been collected from history (or which can be expressed by a user). It also requires thinking about exactly what is missing from the cost model. Do we really care if a locomotive that came in on a particular inbound train is assigned to a particular outbound train? Or is the real issue that incoming eastbound trains put locomotives on a particular track that are best suited for serving outgoing eastbound trains? If so, matching incoming train ID to outgoing train ID is too much detail.

5. Conclusions

This paper presents a modeling framework to solve resource allocation problems with incomplete information characterized by the inability of the modeler to express all operational behavior in terms of cost functions due to missing elements of data. We take advantage of the observation that criticisms of a model can often be expressed as low-dimensional patterns. We use a proximal point framework to develop a function known as the pattern metric for representing patterns in an optimization model. We present the modeling paradigm as combining the "bottom-up" cost function approach with a "top-down" modeling method of incorporating patterns using a quadratic functional form for the pattern metric. Our results using real-world data from a locomotive scheduling operation show that this approach is promising for practical implementation.

Acknowledgement

This research was supported in part by grant AFOSR-FA9550-05-1-0121 from the Air Force Office of Scientific Research and NSF grant CMS-0324380. The authors would also like to acknowledge the helpful and timely comments of two anonymous referees.

References

Ahuja, R. and Orlin, J. (2001) Inverse optimization, *Operations Research* **49**(5), 771–783.

Bertsekas, D. (1995) *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts.

Burton, D., Pulleybank, B. and Toint, P. (1997) The inverse shortest paths problem with upper bounds on shortest paths costs, *in* P. Pardolos, D.W.Hearn and W.H.Hager, eds, 'Network Optimization', Vol. 450, pp. 156–171.

Fourer, R., Gay, D. M. and Kernighan, B. W. (1993) *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, South San Francisco, CA.

Parsons, J. (1996) An information model based on classification theory, *Management Science* **42**(10), 1437–1453.

Rockafellar, T. (1976) Monotone operators and the proximal point algorithm, *SIAM Journal of Control and Optimization* **14**(5), 877–898.

Rudin, W. (1964) *Principles of Mathematical Analysis*, McGraw-Hill Book Company, New York, NY.

Ruszczyński, A. (1987) A linearization method for nonsmooth stochastic programming problems, *Mathematics of Operations Research* **12**(1), 32–49.

Sokkalingam, P., Ahuja, R. and Orlin, J. (1999) Solving inverse spanning tree problems through network flow techniques, *Operations Research* **47**(2), 291–298.

Vanderbei, R. (1999) An interior point code for quadratic programming, *Optimization Methods and Software* **11**(2), 451–484.

Zhang, J. and Liu, Z. (1999) A further study on inverse linear programming problems, *Journal of Computational and Applied Mathematics* **106**(2), 345–359.

Appendix

Our primary goal is the proof of theorem 3.1, which we restate here:

Theorem 3.1. *We have for* $0 < \theta_1 < \theta_2 < \infty$*:*

$$\sum_{\hat{a} \in \hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 < \sum_{\hat{a} \in \hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2,$$

and:

$$C(x^*(\theta_1)) < C(x^*(\theta_2)).$$

We begin with a lemma that states that for $0 \leq \theta < \infty$, the model flows are never identical to the pattern flows.

Downloaded by [Princeton University] at 07:52 04 January 2013

Lemma 1. *We must have $\forall\theta$ such that $0 \leq \theta < \infty$ in the following:*

$$\{x^*_{\hat{a}}(\theta)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \tag{29}$$

Proof. For $\theta = 0$ the statement follows trivially from assumption A.5. For $\theta > 0$ we prove this by showing that if for any positive $\theta' < \infty$ we have:

$$\{x^*_{\hat{a}}(\theta')\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \tag{30}$$

then for $\forall\theta$ such that $0 < \theta < \infty$ we must have:

$$\{x^*_{\hat{a}}(\theta)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \tag{31}$$

We see that the proof then follows trivially. We consider two cases:

Case 1: $\theta > \theta'$: Consider the following optimization problem:

$$x^*(\theta) = \arg\min_{x\in\mathcal{X}} C(x(\theta)) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2$$

We have the following relation:

$$\min_{x\in\mathcal{X}} C(x(\theta)) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 = \min_{x\in\mathcal{X}} C(x(\theta)) + \theta'\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta') - x^p_{\hat{a}}\right\|^2 + (\theta - \theta')\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta') - x^p_{\hat{a}}\right\|^2$$

It follows that:

$$\min_{x\in\mathcal{X}} C(x(\theta)) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 \geq \min_{x\in\mathcal{X}} C(x(\theta)) + \theta'\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 + \min_{x\in\mathcal{X}}(\theta - \theta')\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2 \tag{32}$$

By our assumption, $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is minimizer to the first optimization model on the right hand side. However the second optimization problem:

$$\arg\min_{x\in\mathcal{X}} (\theta - \theta')\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta'') - x^p_{\hat{a}}\right\|^2$$

where $\theta'' = \theta - \theta'$, is the standard projection problem on a convex set $\mathcal{X}$ which by extension of Weierstrass' theorem to strictly convex functions has a unique minimum (Bertsekas, 1995). This unique minimum is clearly $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ since $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is a feasible point. Thus it follows that $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is the optimal solution to the optimization model indicated on the left hand side of the Equation (32), that is, $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ solves the following optimization problem:

$$\min_{x\in\mathcal{X}} C(x(\theta)) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2$$

and thus we prove the statement (31) for the case $\theta > \theta'$. We now consider the other possible situation:

Case 2: $\theta < \theta'$: We have:

$$\min_{x\in\mathcal{X}} C(x(\theta')) + \theta'\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta') - x^p_{\hat{a}}\right\|^2 = \min_{x\in\mathcal{X}} C(x(\theta')) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta') - x^p_{\hat{a}}\right\|^2 + (\theta' - \theta)\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta') - x^p_{\hat{a}}\right\|^2 \tag{33}$$

In this equation $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is minimizer to the optimization problem on the left hand side by our assumption. Again by Weierstrass' theorem $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is the unique optimizer to the optimization model represented by the last term on the right hand side of Equation (33). We also see that the optimization model in a pattern metric is a strictly convex function in $\{x_{\hat{a}}(\theta)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ and Equation (33) is a sum of two strictly convex functions. Thus it follows that $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is the minimizer to the optimization problem:

$$\min_{x\in\mathcal{X}} C(x(\theta)) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2$$

and thus we prove the statement in Equation (31) for the case $\theta < \theta'$. This implies that

$$\hat{x}^{p*} = \lim_{\theta\to 0}\left(\arg\min_{x\in\mathcal{X}} C(x) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2\right) \tag{34}$$

To complete the proof we need to show that the following is true:

$$\hat{x}^{p*} = x(0), \quad \text{for some} \quad x(0) \in \mathcal{X}(0)$$

where $\mathcal{X}(0)$ is the set of alternate optimal solutions to the optimization model: $\min_{x\in\mathcal{X}} C(x)$. Let us assume the following:

$$y^* = \lim_{\theta\to 0}\left(\arg\min_{x\in\mathcal{X}} C(x) + \theta\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\|x_{\hat{a}}(\theta) - x^p_{\hat{a}}\right\|^2\right) \tag{35}$$

The proof that $y^* \in \mathcal{X}(0)$ is given after this proof. From Equation (34) it follows that $\hat{x}^{p*} = \hat{y}^*$. This violates assumption A.4. and hence (31) is false. It is easy to see that this implies statement (29) is true by contradiction. If statement (29) is not true then the premise stated in Equation (30) is true implying the truth of statement (31) which we have already shown cannot hold. ■

Corollary 1. *For any two scaling factors θ_1 and θ_2 such that $0 < \theta_1 < \theta_2 < \infty$ we have the following:*

$$\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \tag{36}$$

Proof. We show Equation (36) by contradiction using the strict convexity property of the optimization model in a pattern metric. Assume there exists $\theta_2 > \theta_1 > 0$ such that $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$. In this case we have $C(x^*(\theta_1)) = C(x^*(\theta_2))$. To see this we note that $x^*(\theta_1)$ is a feasible solution and $x^*(\theta_2)$ is optimal to the model in a pattern framework with θ_2 as the scaling factor and

Downloaded by [Princeton University] at 07:52 04 January 2013

hence:

$$C(x^*(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2$$
$$\leq C(x^*(\theta_1)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2$$

For $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ we see that the above inequality reduces to:

$$C(x^*(\theta_2)) \leq C(x^*(\theta_1))$$

However combining this with the Equation (14) in theorem 3.1 we see that:

$$C(x^*(\theta_2)) = C(x^*(\theta_1)) \tag{37}$$

We have the following equation regarding the model with a pattern metric:

$$\min_{x\in\mathcal{X}} C(x(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 = \min_{x\in\mathcal{X}} C(x(\theta_2))$$
$$+\theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 + (\theta_2 - \theta_1) \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2$$

and $x^*(\theta_1)$ is the solution to the following optimization problem:

$$x^*(\theta_1) = \arg\min_{x\in\mathcal{X}} C(x(\theta_1)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2$$

Under the assumption that $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ and Equation (37) we see that $x^*(\theta_1)$ solves the following optimization problem:

$$\min_{x\in\mathcal{X}} C(x(\theta_2)) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2$$

The above model is also strictly convex in $\{x_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ and hence the minimizer $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} = \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is unique. Thus it follows that $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ minimizes $(\theta_2 - \theta_1)\sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}}(\theta_2 - \theta_1) - x^p_{\hat{a}}\|^2$ over the feasible region $\mathcal{X}$. However the problem:

$$\arg\min_{x\in\mathcal{X}} (\theta_2 - \theta_1) \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta') - x^p_{\hat{a}} \right\|^2$$

where $\theta' = \theta_2 - \theta_1$, has a unique minimum as a consequence of Weierstrass' theorem. This unique minimum is clearly $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ since $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ is a feasible point which contradicts the statement that $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ as implied by Equation (29). Thus we prove $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^*_{\hat{a}}(\theta_2)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ (and consequently $x^*(\theta_1) \neq x^*(\theta_2)$) stated in Equation (36). ■

The following proposition which is a consequence of lemma 1 is required to prove theorem 3.1.

Proposition 4.1. *The function* $|x^*_{\hat{a}\hat{d}}(\theta) - x^{p*}_{\hat{a}\hat{d}}|$ *is strictly monotone in* θ *for* $\theta > 0$ *for all* $\hat{a} \in \hat{\mathcal{A}}$, *for all* $\hat{d} \in \hat{\mathcal{D}}_{\hat{a}}$.

Proof. Consider two scaling factors θ_1 and θ_2 such that $0 < \theta_1 < \theta_2 < \infty$. We have:

$$C(x) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}} - x^p_{\hat{a}} \right\|^2 = C(x) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}} - x^p_{\hat{a}} \right\|^2$$
$$+ (\theta_2 - \theta_1) \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}} - x^p_{\hat{a}} \right\|^2$$

We see that $C(x) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$ is a sum of two strictly convex functions, $C(x) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$ and $(\theta_2 - \theta_1)\sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$. The minimizer of $C(x) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$ is given by $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ which as a consequence of lemma 3.1 has the property $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$. The minimizer of $(\theta_2 - \theta_1)\sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$ is $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$. From convexity theory this implies that the minimizer of $C(x) + \theta_2 \sum_{\hat{a}\in\hat{\mathcal{A}}} \|x_{\hat{a}} - x^p_{\hat{a}}\|^2$ is a convex combination of $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ and $\{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$.

Since $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ and by assumption A.5 this inequality holds elementwise we see that $x^*_{\hat{a}\hat{d}}(\theta_2)$ is a strict convex combination of $x^*_{\hat{a}\hat{d}}(\theta_1)$ and $x^{p*}_{\hat{a}\hat{d}}$, that is, we have for a parameter $\lambda_{\hat{a}\hat{d}} \in (0, 1)$ the following:

$$x^*_{\hat{a}\hat{d}}(\theta_2) = \lambda_{\hat{a}\hat{d}} x^*_{\hat{a}^*\hat{d}^*}(\theta_1) + (1 - \lambda_{\hat{a}\hat{d}}) x^{p*}_{\hat{a}\hat{d}} \quad \forall\hat{a} \in \hat{\mathcal{A}}, \quad \forall\hat{d} \in \hat{\mathcal{D}}_{\hat{a}}$$

This implies that:

$$\left| x^*_{\hat{a}\hat{d}}(\theta_2) - x^{p*}_{\hat{a}\hat{d}} \right| < \left| x^*_{\hat{a}\hat{d}}(\theta_1) - x^{p*}_{\hat{a}\hat{d}} \right| \quad \forall\hat{a} \in \hat{\mathcal{A}}, \quad \forall\hat{d} \in \hat{\mathcal{D}}_{\hat{a}}$$

Since we choose θ_1 and θ_2 arbitrarily and since $\{x^*_{\hat{a}}(\theta_1)\}_{\forall\hat{a}\in\hat{\mathcal{A}}} \neq \{x^{p*}_{\hat{a}}\}_{\forall\hat{a}\in\hat{\mathcal{A}}}$ holds for all $\theta \geq 0$ we see that $\{|x^*_{\hat{a}\hat{d}}(\theta) - x^{p*}_{\hat{a}\hat{d}}|\}_{\forall\hat{a}\in\hat{\mathcal{A}},\forall\hat{d}\in\hat{\mathcal{D}}_{\hat{a}}}$ is a strict monotonic decreasing function in θ, $\theta > 0$. ■

We finally prove theorem 3.1 based on the result of proposition 4.1 which holds under assumption A.5.

Proof. Using proposition 4.1 along with Equation (15) we obtain the strict monotone property stated in Equation (12):

$$\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 < \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2$$

To prove Equation (13) we note that $x^*(\theta_1)$ solves the following minimization problem

$$\min_{x\in\mathcal{X}} C\left(x(\theta_1)\right) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2$$

Since $x^*(\theta_2)$ is a feasible solution we have:

$$C(x^*(\theta_1)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2$$
$$\leq C(x^*(\theta_2)) + \theta_1 \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2$$

Rearranging we get:

$$\theta_1 \left(\sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_1) - x^p_{\hat{a}} \right\|^2 - \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x^*_{\hat{a}}(\theta_2) - x^p_{\hat{a}} \right\|^2 \right)$$
$$\leq C(x^*(\theta_2)) - C(x^*(\theta_1))$$

Downloaded by [Princeton University] at 07:52 04 January 2013

and using the strict monotone property of the pattern metric in Equation (12) it follows that $C(x^*(\theta_1)) < C(x^*(\theta_2))$. ■

Now we show that y^* derived in Equation (35) is in the set $\mathcal{X}(0)$. We first note that the optimization model in a pattern metric given by:

$$C(x,\theta) = C(x) + \theta \sum_{\hat{a}\in\hat{\mathcal{A}}} \left\| x_{\hat{a}}(\theta) - x^p_{\hat{a}} \right\|^2$$

is uniformly continuous over $\mathcal{X}$ since $C(x,\theta)$ is continuous and the feasible region $\mathcal{X}$ is compact (see Rudin, 1964, p. 78). Consider a sequence of strictly decreasing scaling factors $\{\theta_1,\ldots,\theta_k,\ldots\}$ where $\theta_k \to 0$ as $k \to \infty$. Note that $C(x,0) = C(x)$ and as $\theta_k \to 0$, $C(x,\theta_k) \to C(x)$.

Let $\mathcal{Y}_k$ be the set of minimizers of the optimization model $C(x,\theta_k)$. Let $y^*_k \in \mathcal{Y}_k$ be a sequence of minimizers of $C(x,\theta_k)$ with $\theta_k \to 0$ and let:

$$y^* = \lim_{k\to\infty} y^*_k$$

Recall that $\mathcal{X}(0)$ is the set of all optimal solutions to the model: $\min_{x\in\mathcal{X}} C(x)$. We prove the following:

Theorem 4.2. *For any sequence* $\{y^*_k\}_{k=1,2,\ldots}$ *we have:*

$$y^* \in \mathcal{X}(0)$$

Proof. We show the proof by contradiction. Suppose $C(x) < C(y^*)$ where x is any feasible point in $\mathcal{X}$. Let:

$$\epsilon = C(y^*) - C(x)$$

We have the following:

$$\begin{aligned} &C(y^*_k,\theta_k) - C(x,\theta_k) \\ &\quad = [C(y^*_k,\theta_k) - C(y^*_k)] + [C(y^*_k) - C(y^*)] \\ &\qquad + [C(y^*) - C(x)] + [C(x) - C(x,\theta_k)] \end{aligned} \tag{38}$$

Choose K in the set of natural numbers such that $\forall k \geq K$ we have:

$$|C(y^*_k,\theta_k) - C(y^*_k)| \leq \frac{\epsilon}{4}$$

and

$$|C(y^*_k) - C(y^*)| \leq \frac{\epsilon}{4}$$

and

$$|C(x) - C(x,\theta_k)| \leq \frac{\epsilon}{4}$$

This is possible because of the uniform continuity property of $C(x,\theta_k)$ over $\mathcal{X}$ and the limit $\lim_{k\to\infty} C(x,\theta_k) = C(x)$. Thus we can rewrite Equation (38) in the following manner:

$$C(y^*_k,\theta_k) - C(x,\theta_k) > -\frac{\epsilon}{4} - \frac{\epsilon}{4} + \epsilon - \frac{\epsilon}{4} = \frac{\epsilon}{4} > 0$$

This contradicts the fact that y^*_k is the minimizer of $C(x,\theta_k)$. Thus our assumption that there is a feasible $x \in \mathcal{X}$ such that $C(x) < C(y^*)$ is false. This implies that $y^* \in \mathcal{X}(0)$. ■

Biographies

Arun Marar is currently a risk management associate at a multi-strategy hedge fund DKR Capital. Previously, he was in risk management at Amaranth Advisors, a multi-strategy hedge fund and MarketRx, a consulting company specializing in application of OR techniques to pharmaceutical marketing. Prior to starting his Ph.D. in Operations Research and Financial Engineering at Princeton University, he was a Research Engineer at CASTLE Laboratory at Princeton University.

Warren Powell is a professor in the Department of Operations Research and Financial Engineering at Princeton University where he is director of CASTLE Laboratory, which specializes in large-scale stochastic resource allocation problems. An INFORMS Fellow, he is the author or coauthor of over 100 refereed publications, and was twice a finalist in the Franz Edelman competition. His research focuses on approximate dynamic programming and its use for solving large-scale stochastic optimization problems (see www.castlelab.princeton.edu).

Sanjeev Kulkarni is a professor in the Department of Electrical Engineering at Princeton University and is an affiliated faculty member with the Department of Operations Research and Financial Engineering and the Department of Philosophy. He is a Fellow of the IEEE, and his research interests span a variety of areas including statistical pattern recognition, nonparametric estimation, learning and adaptive systems, information theory, wireless networks, and signal/image/video processing.