Interfaces
Vol. 42, No. 1, January–February 2012, pp. 74–84
ISSN 0092-2102 (print) | ISSN 1526-551X (online)

informs®
http://dx.doi.org/10.1287/inte.1110.0618

THE FRANZ EDELMAN AWARD
*Achievement in Operations Research*

# Tax Collections Optimization for New York State

Gerard Miller, Melissa Weatherwax, Timothy Gardinier
Department of Taxation and Finance, State of New York, Albany, New York 12227
{gerard.miller@tax.ny.gov, melissa.weatherwax@tax.ny.gov, timothy.gardinier@tax.ny.gov}

Naoki Abe, Prem Melville, Cezar Pendus, David Jensen
IBM Research, Yorktown Heights, New York 10598
{nabe@us.ibm.com, pmelvil@us.ibm.com, cpendus@us.ibm.com, davjen@us.ibm.com}

Chandan K. Reddy
Department of Computer Science, Wayne State University, Detroit, Michigan 48202,
reddy@cs.wayne.edu

Vince Thomas, James Bennett, Gary Anderson, Brent Cooley
Global Business Services, IBM Corporation, Armonk, New York 10504
{vpthoma@us.ibm.com, jjbennet@us.ibm.com, gfa@us.ibm.com, bcooley@us.ibm.com}

The New York State Department of Taxation and Finance (NYS DTF) collects over $1 billion annually in assessed delinquent taxes. The mission of DTF's Collections and Civil Enforcement Division (CCED) is to increase collections, but to do so in a manner that respects the rights of citizens, by taking actions commensurate with each debtor's situation. CCED must accomplish this in an environment with limited resources. In a collaborative work, NYS DTF, IBM Research, and IBM Global Business Services developed a novel tax collection optimization solution to address this challenge. The operations research-based solution combines data analytics and optimization using the unifying framework of constrained Markov decision processes (C-MDP). The system optimizes the collection actions of agents with respect to maximizing long-term returns, while taking into account the complex dependencies among business needs, resources, and legal constraints. It generates a customized collections policy instead of broad-brush rules, thereby improving both the efficiency and adaptiveness of the collections process. It also enhances and improves the tax agency's ability to administer taxes equitably across the broad scope of individual taxpayers' situations. The system became operational in December 2009; from 2009 to 2010, New York State increased its collections from delinquent revenue by $83 million (8 percent) using the same set of resources. Given a typical annual increase of 2 to 4 percent, the system's expected benefit is approximately $120 to $150 million over a period of three years, far exceeding the initial target of $90 million.

*Key words*: dynamic programming; tax policy; decision support systems; data analysis.

The New York State Department of Taxation and Finance (NYS DTF) collects over $1 billion annually in assessed delinquent taxes. The mission of DTF's Collections and Civil Enforcement Division (CCED) is to increase collections, but to do so in a manner that respects the rights of citizens by taking actions commensurate with each debtor's situation. CCED must accomplish this in an environment with limited resources.

New York State has always been aggressive in applying technology to facilitate collections; however, critical resource decisions about "who will work the cases and when" have essentially relied on manual rules based on gut decisions. The result was a one-size-fits-all approach that was both inefficient and inequitable in treating tax debtors. NYS DTF sought a more flexible decision model that would customize its collection activities to an individual debtor's situation.

In a collaborative work, NYS DTF, IBM Research, and IBM Global Business Services developed a novel tax collections optimization solution to meet this challenge. The solution combines data analytics and optimization using the unifying framework of constrained Markov decision processes (C-MDP). It optimizes collection actions to maximize long-term returns, while considering other complex dependencies.

The resulting system has some notable characteristics. Via the tight coupling of data analytics and optimization in the C-MDP framework, it realizes a level of decision automation that had never been achieved. Because the optimization process is guided by data analytics, the resulting collection policy can adapt to changes in the environment. It also optimizes the collection rules, driving long-term returns, which are critical in this domain because all collections actions (e.g., warrant as prerequisites to levies) do not show immediate rewards. This aspect can free management from relying on manual, sequential workflow rules, which have hindered the optimization of collections processes to date. Although the sequential dependencies between various collections stages are automatically discovered, the system also accepts hard constraints, which are expressed as rules, for use in its optimization process. Hence, the system brings together three principal elements of operations research (OR)—analytics, optimization, and rules—in a novel, coherent, and effective manner, and simultaneously attains their corresponding benefits—adaptability, optimality, and practicality.

The system became operational in December 2009, and the results to date are compelling. New York State increased its delinquent revenues by $83 million (8 percent) using the same set of resources. Assuming CCED's typical annual increase of 2 to 4 percent, the expected additional benefits from the case identification and selection system (CISS) are $120 to $150 million in additional tax revenues over the next three years. This far exceeds the initial target of $99 million. We expect revenue increases to improve as the system adapts to environmental changes. In addition, we have seen an increase in the productivity of our primary enforcement actions and a decrease in the need to use them. The use of detailed characteristics to make informed decisions about individual tax debtors allows us to treat taxpayers as individuals. The result has been both an increase in revenues and a decrease in the number of enforcement actions—a situation in which all parties win.

## Challenges

The NYS Tax Department has always stayed abreast of and taken advantage of new technologies that could improve its collections efficiency. In 2005, the department started discussions with IBM about an OR project, which we called the Case Identification and Selection System (CISS). During the previous 10 years, the tax department had implemented numerous changes to modernize its operations; however, problems still existed. To understand the value that OR has added through CISS, an understanding of the background and nature of the task of collecting delinquent taxes in New York State is important.

Tax collection in New York State, much like everything else, is made more complex by the number and diversity of its citizens. Tax debtors come in all shapes, sizes, and levels of sophistication; some are more willing to pay their taxes than others. Collection techniques appropriate for collecting a debt from a small family-owned business are unlikely to be the best techniques for collecting taxes from a Fortune 500 company; however, both are in our debtor inventory. In addition, in New York, unlike some other states, the collections division is responsible for collecting some 40 state and local taxes.

The laws dictating how tax debts are collected in New York State are an additional hurdle. The state must first file an administrative lien, also known as a tax warrant or judgment, before any levy, wage garnishment, or seizure of other assets is allowed. This requirement made it necessary for our OR solution to address not only what was the best decision at a point in time, but to anticipate the future impact that a decision would have (i.e., a levy requires a prior warrant).

In the decade prior to 2005, CCED operated as a field-based collections operation that relied almost exclusively on effective and costly face-to-face contact with tax debtors at their homes or businesses. By 2005, CCED had centralized its operations around a state-of-the-art call center and a reduced field staff that focused primarily on business tax debts. It reduced its work force by 30 percent and significantly shifted its staff from field to call center functions. These adjustments and the corresponding automation of many tasks resulted in an increase in collections revenue to twice what it had been in 1995 with no significant increase in the amount available for active collections. In 1995, CCED collected $500 million; by 2005, its collections were just under $1 billion although it had reduced its staff by 30 percent.

The success story was good; however, problems still existed. As CCED evaluated where it had come from and where it wanted to go, one fact became clear. Decisions on what collections path a case should take were still being made based on crude formulas developed based on that most basic analytical tool—the gut. Initial routing choices—should a particular case be worked by call center staff, field staff, specialized units, or not at all—were based on broad linear statements such as "business tax cases over $*x* will go to the field." The problems with this are obvious. Although some business tax cases over $*x* are best suited for the field, many could be collected successfully by telephone. The cost of a telephone call is dramatically less than a field visit. Additionally, cases stayed in the call center too long and were not being assigned to the field fast enough. By the time a case was assigned to the field, the taxpayer was often out of business and the debt no longer collectible.

This one-size-fits-all approach was also problematic when looked at relative to the collection cycle within the call center. Once assigned to call center inventory, all cases followed a linear path of allowable actions that were based on law and policy. All cases started with an initial telephone contact in which we would try to come to an agreement with the taxpayer to avoid any civil enforcement actions. If left unresolved, the case proceeded to warrant and, if still unresolved, to levy or wage garnishment based on the availability of those sources. All tax debtors were treated the same in this process. Decisions to take action were based on what we could do. Via the magic of OR, CISS now allows us to make decisions based on what we should do.

Let's look at that last statement: decisions based on what we should do. Why is that so important? In evaluating the process that we describe above, does it make sense to treat all tax debtors the same? Not only was our linear, gut-driven process not maximizing revenue, it was also unfair to taxpayers. If the tax debtor is a chronic delinquent who has required enforcement action in the past, is a telephone call likely to result in payment? Conversely, if this is the first infraction for a tax debtor who has a long history of voluntary compliance, should we not take extra efforts to contact the debtor before we proceed with warrant and levy procedures?

We needed a way to make decisions based on the individual characteristics of each tax debtor. The linear one-size-fits-all approach was both inefficient and unfair; there had to be a better way. How could we possibly find a way to do this on an inventory of more than a million tax debtors, all with their own individual stories? Our answer was OR and business analytics.

## Solution

### Solutions Considered

Because of the complexity of the collection process, and particularly because of the legal and business constraints, following rigid manually constructed rules to guide collection activities is common practice. Even in state-of-the-art rule-based collection systems, the role of data analytics is typically limited to that of augmenting the rules engine with scores that can be referenced by the rules. However, existing systems based on manually constructed rules suffer from the shortcoming that the rules do not adapt to changes in the environment, including economic and organizational changes. Hence, the need for a more adaptive alternative approach has been recognized in the field.

An alternative to such a rigid manual rule system is a mathematical model that describes the complex collections process, applies analytics to estimate it, and then optimizes to arrive at a collections policy within the class of policies describable in the model. The MDP provides one such framework with a rich-enough vocabulary to describe the complexity involved in a collections process (e.g., the presence of different stages and the intricate interactions among them). An earlier work at General Electric applied an MDP-based approach to the problem of managing consumer credit delinquency (Makuch et al. 1992). However, this work was based primarily on optimization techniques and did not integrate data analytics and optimization to the extent that we did in the present work. Specifically, the states of the MDP were fixed a priori, and the transition probabilities among them were estimated prior to the optimization process. Today, with the abundance of historical data in digital form and with advances in data mining and analytics technology, the business requirements and expectations demand a collections management system that can customize enforcement actions depending on a diverse set of individual taxpayer characteristics. Working with a small fixed number of taxpayer states is insufficient for this.

As we discuss in this paper, we developed a novel tax collections optimization solution that uses a unique combination of data analytics and optimization techniques in a unified manner, based on the framework of C-MDP. The system optimizes the collection actions of agents with respect to maximizing long-term returns, while taking into account the complex dependencies among business needs, resources, and legal constraints.

With C-MDP and the associated methodology of constrained reinforcement learning (C-RL), the system realizes a level of decision automation on a scale that had never been achieved in real-world business optimization applications. In addition, to deal with the complex constraints and dependencies that have hindered the automation of collections decision processes, it also accepts constraints expressed as rules. Thus, analytics, optimization, and rules are brought together in a coherent manner to provide a practical and viable solution.

### Solution Description

In this subsection, we describe our solution in detail. The description is at a conceptual level that draws on concrete examples. We refer the reader to Abe et al. (2010) for the technical details.

As we mention above, we use the MDP framework (Puterman 1994, Tsitsiklis and Van Roy 1997) to formulate the process of tax collections. Intuitively, the states of the MDP represent the stages in the collection process that respective taxpayers' cases undergo. The actions are the collections actions taken by DTF for those cases, and the rewards are the dollars collected, optionally reflecting other measures of compliance (see Figure 1). We assume that the taxpayers make transitions through this state space representing the collections process, and their transition probabilities and their rewards will naturally depend on the choice of the collections actions. The goal of our solution is to find a near-optimal policy that determines the appropriate choice of collections actions to take, depending on the taxpayer's current state.

An important point worth emphasizing is that although Figure 1 appears to show relatively few states, which are fixed a priori, this is not the case. We elect to represent the state space by a high-dimensional feature space that consists of hundreds of features summarizing each taxpayer's history, including transactions and collection actions taken. We used approximately 200 features in the deployed system; however, the system can accommodate a variable number of features. Partitions (segments) of this

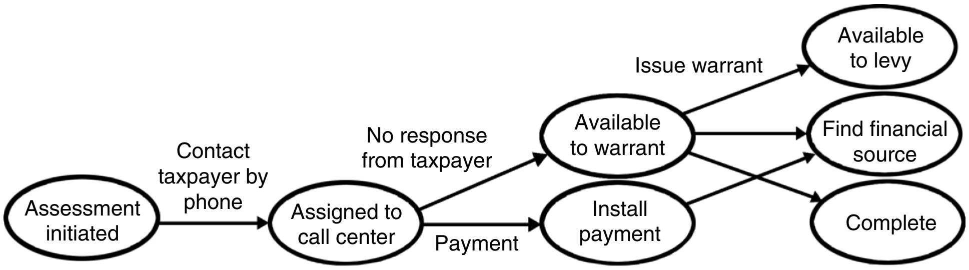


Figure 1: The graphic illustrates an MDP formulation of the collections process.

feature space are automatically discovered within our modeling and optimization procedure.

When we can reasonably assume that the full structure of the MDP is known or is estimated a priori (i.e., we can reliably estimate the state set and the transition and expected reward functions), then well-known methods (e.g., the value iteration procedure) exist to solve for an optimal policy in the MDP. However, we emphasize that this assumption is not reasonable for a large-scale real-world problem such as the one we have at hand. Given that we have a high-dimensional feature space representing the state space, reliably estimating the structure of the MDP is a formidable task. Therefore, we turn to the approach known as RL (i.e., approximate dynamic programming) or the collection of methods for learning a near-optimal policy in an MDP, given access only to the data generated from the MDP, not the MDP structure itself (Kaelbling et al. 1996, Puterman 1994, Sutton and Barto 1998, Tsitsiklis and Van Roy 1997).

Another important aspect of our application is the existence of various constraints that must be respected in our search for an optimal collections policy. Several types of constraints are involved. The first category consists of resource constraints that are imposed because some collections actions consume physical resources. Other constraints might not be resource related; these include bounds on the number of certain types of actions that are considered reasonable in a given period. The second major category of constraints, called action constraints, are those that are imposed by business or legal considerations.

The foregoing considerations motivate our solution to be based on a constrained version of RL methodology (C-RL). Although the literature includes a considerable amount of prior work in the area of C-MDP (Altman 1992), less is known in the area of C-RL. Existing methods within the C-MDP literature, which are primarily based on optimization, are not well-suited to our current formulation, because they do not readily render themselves to be combined effectively with statistical data modeling, except through direct estimation of the C-MDP structure separately from the optimization process. In this work, we elect an integrated approach in which data analytics and optimization are tightly coupled and applied in an iterative fashion.

Figure 3 (see the appendix) provides an informal description of how the generic method works. The key lies in the constrained version of Bellman's equation, shown in the center of the figure, for estimating the expected long-term reward as a function of state and action. The concept is that the long-term reward ($R$ on the left side) equals the sum of the immediate reward and the long-term reward expected from the next state.

As in the original value iteration procedure for solving Bellman's equation, we iterate this process each time by updating the value of $R$ on the left side using the estimated value of $R$ from the previous iteration on the right side. Note that the equality corresponds to regression or estimation, whereas maximization corresponds to optimization, both over a large population. In our method, we do the estimation step using a scalable procedure for segmented linear regression; in each segment, $R$ is estimated using regression in terms of action variables (Apte et al. 2001, Natarajan and Pednault 2002). Thus, the procedure discovers segments in the feature space, each of which is relatively uniform with respect to the effects of actions. When this process is completed, the output (a set of action effectiveness coefficients per segment) is entered into the optimization process, which determines a policy that allocates actions to maximize $R$ over the population while respecting various constraints. This iterative procedure, consisting of analytics and optimization, is repeated many times; each time, it performs further "looks ahead" and improves the quality of estimation and optimization.

The actual algorithm we deploy is more involved; it includes steps to normalize $R$ in its iterative process, which brings out the relative effects of competing actions on $R$. However, the basic concept is captured in this intuitive account. Abe et al. (2010) provide details.

The quality of the models improves through this iterative process. This includes the segmentation. The number of segments is generally different in each iteration, and the quality of segmentation improves as the iterations proceed, as do the overall action allocations. For example, note that the segmentation in the initial iteration is optimized with respect to the estimation of immediate rewards, which is clearly suboptimal for estimating the long-term reward. This is also confirmed in empirical evaluation using multiple data sets, as Abe et al. (2010) discuss.

The discovered segments and the action allocations given to them by the constrained optimization procedure form the collections rules. Note also that the action constraints can involve complex definitions; hence, representing them as rules is natural. Thus, the developed system can be viewed as an automatic rule generation system, which receives as input action constraint rules (and training data), invokes analytics and optimization, and generates collections rules. Both the input and output rules of the engine can be housed in a rules management system, with the engine providing an automatic rule generation capability.

Another key advantage of the engine is its focus on maximizing long-term rewards. By using the C-MDP framework, the rules it generates are optimized with respect to the long-term return, which is critical in the collections domain because all collections actions do not enjoy immediate rewards (e.g., warrants as prerequisites to levy, as noted earlier), and looking ahead is necessary to fairly assess the value of such actions. Rules optimized from a myopic viewpoint (i.e., with respect to immediate reward maximization) would not be suitable. Although the generated rules are optimized with respect to long-term rewards, they are instance-in-time rules that specify which actions are to be taken now, and they can be executed in a rules engine. This is an added advantage of our MDP formulation.

Scalability is another strength of our solution. As its components do, it uses a scalable segmented regression engine (Apte et al. 2001, Natarajan and Pednault 2002) and a fast linear program solver (COIN-OR), both of which were developed primarily at IBM Research. These modules are incorporated in a unifying framework of C-RL; the solution provides a scalable system that is well suited for real-world deployments.

## Deployment

In this section, we give some concrete examples of the key artifacts in the system deployed at NYS DTF. These include actions, resource constraints, state space (modeling) features, legal and business constraints, output segments, and action allocations.

### Actions and Resource Constraints

In the legacy system, the overall collections process is guided by the notion of case assignment. We have accordingly designed the set of actions: we consider direct collections actions that can be taken at an assigned organization, and indirect or movement actions that move cases from one organization to another. Table 1 lists the set of actions being considered. The first and third groups consist of direct actions, and the second group includes the movement actions. The third group (i.e., perform field visit) is special in that it can only be performed in district offices.

Resources are equated to the person-hours available in various organizations for performing the collections actions being considered. For each action that we consider, we estimate the expected amount of time required to perform it based on historical data (see Table 1, column 2); we use these hours in conjunction with the person-hours in organizations to determine the resource constraints. Note that direct actions consume resources from the organization to which the case is assigned currently, whereas indirect actions, in and of themselves, do not consume any resources. However, they will incur resources in their future organizations; we consider this in evaluating the lookahead mechanism inherent in the C-MDP framework. Additionally, we provide a form of constraints to

| Action | Hours | Upper bound |
|---|---|---|
| Collections actions | | |
| Contact taxpayer by mail | 0.01 | 5,000 |
| Contact taxpayer by telephone | 0.14 | 2,000 |
| Create warrant | 0.01 | 5,000 |
| Create income execution | 0.01 | 10 |
| Create levy | 0.09 | 5,000 |
| Movement actions | | |
| Move to district office | 0 | 5,910 |
| Move to high-value team | 0 | 330 |
| Move to collection vendors | 0 | 340 |
| Move to indiv. case enforcement | 0 | 100 |
| Organization-specific action | | |
| Perform field visit | 0.625 | 5,000 |
| No action | | |
| Take no action | 0 | 100,000 |

**Table 1: The table illustrates collections actions being considered, including the hours required to perform them and upper bounds on their allocations specified in hours per day.**

explicitly control the number of allocations of actions that we call upper bounds. Because there is one unified action (i.e., move to district office) for moving a case to any district office, unlike other specialized organizations, we allow for an additional type of bound per district office.

### Modeling Features

Given data consisting of taxpayer background information, complete taxpayer history of transactions (i.e., payments) and actions (e.g., contact and collections actions) taken onto them by the DTF, we generate a time-stamped sequence of feature vectors at multiple sampling (or evaluation) time steps for each taxpayer case; we use these as training data in the C-RL procedure. In the deployed system, we used approximately 200 modeling features. Below, we list some examples that we have grouped into categories based on their features.

(1) Taxpayer: number of nonrestricted financial sources, sales tax inactive indicator, number of bankruptcy filings;

(2) liability: total liability balance, sum of collectible assessments, sum of assessments available to warrant;

(3) transactional: tax paid previous year, number of payments since last action, number of payments to date, sum of payments during previous year;

(4) collections: number of open perfected warrants, days since last warrant perfected.

### Legal and Business Constraints

In addition to the modeling features, the engine uses a group of features called action constraints—binary features specifying, for each action considered, whether the action is currently allowed on the case, according to the business and legal rules. The generation of these features is done by entering the modeling feature vectors into the rules engine, which contains a catalog of approximately 300 business and legal rules that users have constructed (see Table 2).

| Rule no. | Rule contents |
|---|---|
| Contact rule | |
| 502.12 | A collection letter should not be sent to a taxpayer whose mailing address is invalid. |
| 2,000.1 | A contact action should only occur for a taxpayer with at least one open mature assessment. |
| 2,005.9 | A contact by mail must not be made for a taxpayer with an active promise to pay within 30 days. |
| Levy rule | |
| 2,601 | A levy is not allowed for a taxpayer unless the taxpayer has at least one perfected warrant. |

**Table 2: The table shows samples of action constraint rules.**

### Output Segments and Allocations

Table 3 shows an example segment output by the engine. Some interesting observations can be made. Notice that this segment (i.e., Segment 212), for which many warrant actions are allocated, includes the condition that the “number of nonrestricted financial sources” is at least 1. This is one of the conditions necessary for a levy; its inclusion for a segment for which warrant is recommended suggests that the look-ahead mechanism of our RL method is working properly. Also note that “state equals call_center_not_warranted," which means that the case has not been warranted.

| Segment definition | Action allocation |
|---|---|
| Segment 212 | |
| state = call_center_not_warranted | 5,103 crt_warrant |
| and tax_paid_last_year ≤ $X | 152 no_action |
| and 1 ≤ number_nonrestricted_financial_sources | |
| and 1 ≤ sales_tax_inactive_indicator | |
| and number_payments_since_last_action ≤ 1 | |
| etc… | |

**Table 3: The table illustrates an example of an output model segment.**

### Deployment Architecture

The architecture of the deployed system, CISS for Collections, is schematically depicted in Figure 2. The overall solution is deployed and executed within a WebSphere process server (WPS) application server. Data sources include the collections data, tax return data, and additional external data (indicated as System 1, System 2, and System 3 in the figure), which are used to compute summary information about the taxpayers; this information is stored as an intermediate representation referred to as taxpayer profiles. The taxpayer profile exists in a DB2 database (see TP profile in Figure 2) as a single XML document for each taxpayer. The profile records the history of the taxpayer states using the temporal data model approach

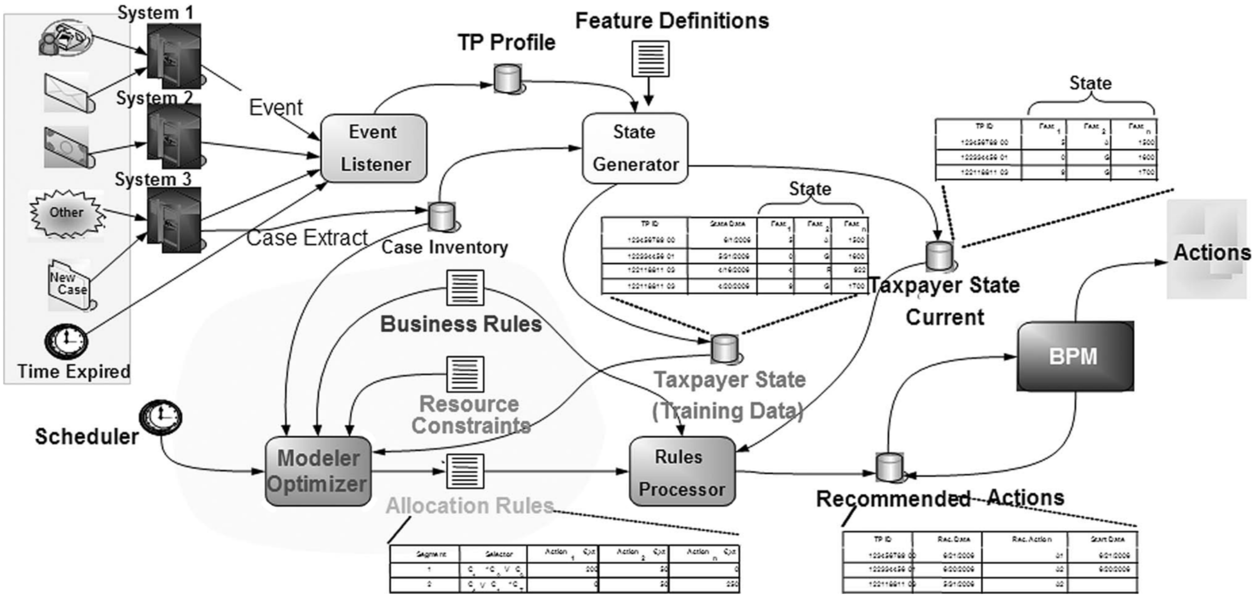


Figure 2: The diagram shows the collections system architecture.

suggested by Wang and Zaniolo (2004), using transactional semantics with a granularity of one day. A simple transform can generate the taxpayer's state as it was on any particular day.

Numerous legacy systems generate events, which are passed to the engine by a separate workflow process that specifies how events flow through the engine. Each event has an XSLT2 (i.e., a language for transforming XML documents into other XML documents) transform that updates the taxpayer profile with the new information. Each week, taxpayers who have had an event, and those whose next review will become due that week, are considered for scoring. They pass through a number of transforms that check their suitability for scoring, apply prescriptive rules, and generate the state features, thus providing the scoring data. An analogous process is used to generate the training data of the C-RL procedure, except in the latter case in which a sequence of state feature vectors per taxpayer is generated at a number of evaluation points in the past. The two types of state feature data are indicated as taxpayer state (current) and taxpayer state (training data), respectively, in Figure 2. Taxpayer profiles are computed for over two million taxpayers, typically as training data for modeling. We use feature data for a subset of them—approximately a few hundred thousand taxpayers, each with five to 10 state feature vectors; on average, they total approximately one million data records.

The overall CISS for Collections solution includes business process management (BPM) technology for monitoring purposes. The automation and human workflow components of the WPS are the cornerstone of the BPM aspects of the CISS for Collections solution. These components help orchestrate over 30,000 automated collections actions and deliver over 8,000 pieces of collection work requiring human involvement each week.

## Benefits

Increasing revenue from collections on outstanding assessed tax liabilities is our project's primary goal. At the end of the first year, we clearly achieved this goal and exceeded expectations. The projected increase in revenue for CISS was $99 million to be achieved over three years. As previously noted, CCED implemented the CISS system in December 2009. A comparison of CCED revenue in 2010 (after the CISS implementation) and in 2009 (prior to CISS implementation) shows an increase of $83 million (8.22 percent). CCED has averaged year-to-year increases ranging between 2–4 percent during the past several years; this places the revenue increase from CISS in its first year in the range of $40–$50 million already realized, with a promise of $120–$150 million over the three-year period originally projected.

This increase was possible despite several conditions that normally would have led CCED to expect reduced collections (or average collections, at best) in 2010 versus 2009. Primary among these reasons was that 2009 had been an unusually successful year. The 6 percent increase over 2008 was the largest year-to-year increase on a steady path of year-to-year progression over the previous five years. CCED expected this because it had implemented a new program that required all New York State banks to match their account holders to judgment debtor files. The result was a bonanza of new levy sources and a significant increase in levy collections. Levy revenue accounted for almost one-third of the total increase in 2009. CCED anticipated that the effectiveness of the matching process would decrease over time. The 8 percent increase over what had been its best year, and without the flood of new sources provided by the first matching in 2009, indicates that some factor had changed.

CCED was also concerned about the effect the economy, which was beginning to show signs of slow recovery in mid-to-late 2009, would have on collections in 2010. We knew that the most recent previous recession in 2001 had caused no real impact in the first year of recovery (2002); however, it resulted in a definite decrease in collections by the second year (2003), in which CCED experienced its only year-to-year decrease in collections in the past decade. Finally, we note that because of budget constraints, maintaining staffing levels was difficult in 2010. In 2009, we were able to maintain a consistent staffing level throughout the year; however, in 2010, we were unable to replace staff members as they left; our staff numbers, particularly in our call center, did not return to expected levels until October 2010.

Because of these concerns (i.e., the prior year's unusually high revenue, the economy, and staffing reductions), CCED had projected only a 3 percent increase over 2009 for 2010. This projected increase consisted almost totally of the $30 million projected for CISS in its first year. The actual increase of over 8 percent was a surprise. In addition to the strong revenue numbers in spite of the obstacles noted, the improvement in dollars collected per action is another strong indication that CISS recommendations for enforcement action were both superior and the root cause of our increased revenue in 2010. Warrants, which as described previously are tax liens against the assets of an individual or business, and levies, which are legal notices that require a third party such as bank to turn over funds to pay a tax debt, account for over $250 and $150 million, respectively, in revenue per year. A comparison of actions taken prior to CISS recommendations with those taken as a result of the recommendations shows a marked increase in revenue per action taken. In the case of warrants, 22 percent more revenue per action taken was achieved; the increase in dollars per levy served was 11 percent.

As we noted in the *Challenges* section, our goals were to both increase collections and to make decisions based on what we should do—not what we could do. Actions such as warrants and levies clearly have serious consequences for tax debtors and impact others who depend on those debtors. If greater amounts of revenue can be collected and CCED reduces the need to use these disruptive collection tools, all parties win.

It had long been a general principle that the more warrants we did and the more levies we served, the more money we would collect. CISS forced us to reevaluate that concept. We reduced the number of warrants by 9 percent and the number of levies served by 3 percent, while increasing revenue from both in 2010. This translates to thousands of New York State citizens who did not have to undergo the ordeal of finding out that money had been taken from their bank accounts or their credit had been harmed. CISS makes the recommendation to take these actions based on the tax debtor's prior history and of what will work now—not what action is next in a linear process.

The more-complex tax debtor cases require more than a point-in-time recommendation. These cases must be separated from those that can be worked appropriately via a series of recommendations for action as soon as possible. They must be assigned to a field agent and are often time sensitive because the debtor might go out of business. CISS has reduced the average age of a case that has been assigned to a field office by 9.3 percent. An evaluation of collections by our field operations shows this has paid off. Revenue from field operations increased by $32 million (12 percent) in 2010 compared to 2009.

## Appendix. The Core Algorithm

In this appendix, we briefly describe the core C-RL algorithm used in our solution. Recall that the objective associated with an MDP is to find a policy, $\pi$, that maximizes the expected cumulative reward,

$$E\left[\sum_{t=0}^{\infty} R(s_t, \pi(s_t))\right],$$

where $s_t, t = 0, \ldots, \infty$, are the sequence of states resulting from the repeated application of the policy.

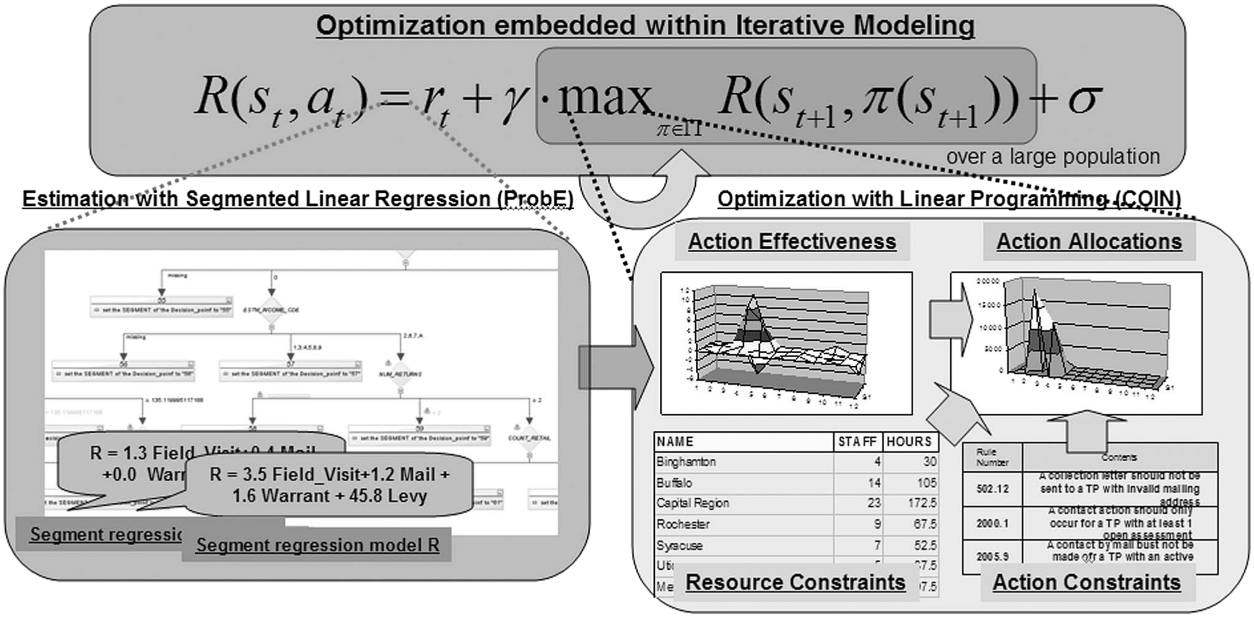
As we mentioned earlier, the key concept behind the class of C-RL methods we use in this work is in the constrained version of Bellman's equation, in which the maximization on the right side is obtained through constrained optimization rather than through strict maximization (see Figure 2). Loosely based on this idea, we can derive constrained versions for many known RL methods, including constrained Q-learning (Watkins 1989). Although the foregoing discussion motivates a family of methods, not all of them are well suited to real-world applications. Dealing with variable time intervals between actions is a critical issue. Among the body of past works that addressed the problem of extending Q-learning and other related learning methods to variable time intervals and continuous-time setting (Baird 1994), Baird's advantage updating algorithm is particularly attractive and has proven effective in previous applications (Abe et al. 2004).

Advantage updating is based on the notion of advantage of an action $a$ relative to the optimal action at a given state $s$, written $A(s, a)$:

$$A(s, a) = \frac{1}{\Delta t_s}(Q(s, a) - \max_{a'} Q(s, a')). \qquad (1)$$

In the above equation, $Q(s, a)$ denotes the Q-function or the two-place value function, and $\Delta t_s$ denotes the time interval between the state $s$ and the subsequent state. The notion of advantage is useful because it normalizes the dependence of the value function on the time interval (by division by $\Delta t_s$), and factors out the influence of the state (by subtraction of $\max_{a'} Q(s, a')$).

Given this notion of advantage, advantage updating is an online learning method that learns this function


Optimization embedded within Iterative Modeling
$R(s_t, a_t) = r_t + \gamma \cdot \max_{\pi \in \Pi} R(s_{t+1}, \pi(s_{t+1})) + \sigma$
over a large population
Estimation with Segmented Linear Regression (ProbE)
Optimization with Linear Programming (COIN)
R = 1.3 Field_Visit+0.4 Mail +0.0 Warr
R = 3.5 Field_Visit+1.2 Mail + 1.6 Warrant + 45.8 Levy
Segment regressic
Segment regression model R
Action Effectiveness
Action Allocations
Resource Constraints
Action Constraints

| NAME | STAFF | HOURS |
|---|---|---|
| Binghamton | 4 | 30 |
| Buffalo | 14 | 105 |
| Capital Region | 23 | 172.5 |
| Rochester | 9 | 67.5 |
| Syracuse | 7 | 52.5 |
| Uti | | 37.5 |
| Me | | 97.5 |

| Rule Number | Contents |
|---|---|
| 502.12 | A collection letter should not be sent to a TP with invalid mailing address |
| 2000.1 | A contact action should only occur for a TP with at least 1 open assessment |
| 2005.9 | A contact by mail bust not be made ofr a TP with an active |

Figure 3: The schematic shows a representation of a generic C-RL procedure.

iteratively, by a coupled set of update rules for the estimates of $A$ and $V$ (the value function), and a normalization step for $A(s, a)$, which drives $\max_{a'} A(s, a')$ toward zero. Although it differs from the canonical Q-learning method, its central step still involves choosing an action that maximizes the $A$-value estimate. Therefore, given the standard version of this algorithm, its constrained version can be derived by replacing the maximization by the appropriate constrained optimization, giving rise to a specific instantiation of the general idea described in Figure 3. Abe et al. (2010) provide details of this method.

## References

Abe, N., N. Verma, C. Apte, R. Schroko. 2004. Cross channel optimized marketing by reinforcement learning. *Proc. 10th SIGKDD Internat. Conf. Knowledge Discovery Data Mining*, ACM, New York, 767–772.

Abe, N., P. Melville, C. Pendus, C. K. Reddy, D. L. Jensen, V. P. Thomas, J. J. Bennett, et al. 2010. Optimizing debt collections using constrained reinforcement learning. *Proc. 16th SIGKDD Internat. Conf. Knowledge Discovery Data Mining*, ACM, New York, 75–84.

Altman, E. 1992. Asymptotic properties of constrained Markov decision processes. Technical Report RR-1598, Institut National de Recherche en Informatique et en Automatique (INRIA), Valbonne, France.

Apte, C., E. Bibelnieks, R. Natarajan, E. Pednault, F. Tipu, D. Campbell, B. Nelson. 2001. Segmentation-based modeling for advanced targeted marketing. *Proc. 7th SIGKDD Internat. Conf. Knowledge Discovery Data Mining*, ACM, New York, 408–413.

Baird, L. C. 1994. Reinforcement learning in continuous time: Advantage updating. *Proc. Internat. Conf. Neural Networks, Orlando, FL*, 2448–2453.

Kaelbling, L. P., M. L. Littman, A. W. Moore. 1996. Reinforcement learning: A survey. *J. Artificial Intelligence Res.* **4**(1) 237–285.

Makuch, W. M., J. L. Dodge, J. G. Ecker, D. C. Granfors, G. J. Hahn. 1992. Managing consumer credit delinquency in the US economy: A multi-billion dollar management science application. *Interfaces* **22**(1) 90–109.

Natarajan, R., E. P. D. Pednault. 2002. Segmented regression estimators for massive data sets. *Proc. 2nd SIAM Internat. Conf. Data Mining*, SIAM, Philadelphia, 566–582.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York.

Sutton, R. S., A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tsitsiklis, J. N., B. Van Roy. 1997. An analysis of temporal difference learning with function approximation. *IEEE Trans. Automatic Control* **42**(5) 674–690.

Wang, F., C. Zaniolo. 2004. Xbit: An XML-based bitemporal data model. *Proc. 23rd Internat. Conf. Conceptual Modeling*, Shanghai, China, Springer-Verlag, Berlin, 810–824.

Watkins, C. J. C. H. 1989. Learning from delayed rewards. Ph.D. thesis, Cambridge University, Cambridge, UK.