Understanding the Performance of Decision Strategies in Dynamic Environments

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Georg Meyer

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Paul E. Johnson

Gediminas Adomavicius

August 2012

© Georg Meyer 2012

Acknowledgements

I want to thank:

My advisors, Paul E. Johnson and Gedas Adomavicius, for their support and constructive criticisms, for many hours of thought-provoking conversations, and for keeping me on track while ensuring quality work;

My committee, Maria Gini, Eugene Borgida, and Alok Gupta, for their insights, feedback, and support during this process, as well as their patience and accommodation when scheduling meetings and defenses;

Patrick O'Connor and JoAnn Sperl-Hillen for their valuable insights into the world of clinicians, their participation and feedback at presentations, their efforts to bring results to the medical community, and their great senses of humor;

Bill Rush, for his encouragement and advice about graduate-student life, and for sharing his advanced knowledge of research and statistical methods;

George Biltz, for interesting discussions and our collaboration to understand exotic analytical methods;

Julian Wolfson, for contributing his knowledge and his ability to convey biostatistical methods and concepts clearly;

Greg Ramsey, Ryan McCabe, Mohamed Elidrisi, and Sunayan Bandyopadhyay for collaboration, inspiration, and commiseration;

Elizabeth Lenzen, Kate Nelson, Noelle French, Julie Cutting, Melissa Grass, Sharon Lamkin, and Earlene Bronson, for friendship, administrative support, and keeping me well-caffeinated;

And, last but not least, the families Krawietz, Meyer, and Preston for their support, curiosity, and questions that forced me to think even harder about how to make my research clear and understandable.

Dedication

To Megan and «The Programme»: L. S. Wang, Miguel Velasco, and Trey Hickman, who kept me grounded and sane.

Table of Contents

List of Tables

List of Figures

1. Background

1.1. Dynamic Decision Making

Edwards (1962) lists the following criteria for a decision making environment to be considered dynamic: (a) a series of decisions is required to reach a goal; (b) the decisions are path-dependent; and (c) the state of the decision problem changes as a result of actions as well as on its own. In addition, dynamic decisions are constrained by time, as seen in emergency management and patient care, for example (Brehmer 1992).

All but the simplest of dynamic decision problems have not been found to be tractable to the analytical normative-descriptive approach which yielded insights into static decision making by comparing decision maker performance to optima derived from theory (Rapoport 1975). In dynamic contexts, decision maker performance is impaired by the presence of delay of feedback, the inability to form a strong mental model of a system's inner workings, and the need to balance competing goals (Funke 1991). When delay of feedback is present, decision makers tend to apply more actions than necessary, which results in overburdening the system after the delay expires (Forrester 1971; Sterman 1989). Opaqueness of a system, which makes it impossible for a decision maker to examine the inner structure of a decision problem, has also been shown to limit performance in dynamic contexts (Brehmer 1992). The complexity introduced by multiple competing goals is illustrated using the example of chronic disease care, in which one goal is to reduce the risk of heart attacks, another is to reduce the risk of

kidney failure, and yet another is to keep the cost of treatment manageable. Because of the inherent complexity of the context, an action may positively affect some aspects but negatively affect others in the long term without a direct indication (feedback) to the decision maker.

Recent research in dynamic decision making has generally focused on understanding cognitive aspects of decision maker performance, such as understanding why decision makers tend to perform poorly in some dynamic environments but succeed in others, and how performance can be improved (Atkins et al. 2002; Gonzalez et al. 2005; Gonzalez et al. 2003; Ramsey 2010). For example, in the domain of information systems, the focus has been on designing systems that free up cognitive (attentional) resources (Lerch & Harter 2001). These studies represent efforts to understand and affect the inner environment of a decision making agent, i.e., the decision making process itself. However, one does not necessarily need to understand the inner environment when behavior is governed by constraints imposed by the external environment (Simon 1969). Decision strategies, such as an organization's guidelines, policies and procedures, represent such constraints. Constraints can make agents more powerful by serving as "scaffolding" that enables agents to build solutions upon what exists in the external environment rather than to solve a problem from scratch (Clark 1998). In fact, it has been shown that decision makers tend to adopt decision strategies when faced with repeated decision problems (Beach & Mitchell 1978; Simon 1969).

Decision strategies take the form of heuristics, guidelines, or algorithms. An example of a heuristic is "choose in round robin fashion" (first choose a, next time choose b, then choose c, then start over with a). This is a 1/N or diversification heuristic (Gigerenzer & Todd 1999). Examples of guidelines are the treatment guidelines published by the Institute for Clinical Systems Improvement (2009). An example of an algorithmic strategy is a trading algorithm used to make investment choices in a market context (B. Johnson 2010). Guidelines may represent some constraints but remain open for interpretation, while algorithms must be fully and explicitly specified. In other words, an algorithm must produce a resulting action for each state it encounters, while a guideline may inform the decision maker but not prescribe an action for every possible state. These categories are not mutually exclusive; heuristics and guidelines can be specified as algorithms if their ambiguities are resolved.

The research proposed here focuses on questions of what decision strategies must be like to succeed or fail in a specific context and how to construct strategies that are effective at producing desired outcomes. To investigate these questions, an appropriate framework to think about decision strategies more formally is needed. Existing literature has converged on a view of decision making as process control and control theory as a suitable and powerful framework for understanding the agent/environment interaction in a dynamic decision problem.

1.2. Dynamic Decision Making as Process Control

A decision strategy controls the decisions in order to achieve goals for a given dynamic decision task. Considering dynamic decision making as a discrete process, the decision strategy can be represented as a *control strategy*. Rapoport (1975), Mackinnon and Wearing (1985) and Broadbent et al. (1986) have all proposed that using control theory from engineering as a model may provide useful perspectives for the study of dynamic decision making. The control theory framework allows us to bring powerful findings from process control to bear on the dynamic decision making problem, specifically the idea of inverse model and forward models, the Good Regulator theorem, and feedback vs. feedforward control.

For successful control, a decision strategy must answer two questions at every *decision point*: (a) What action should be taken? (b) What is the effect of a given action? (Gibson et al. 1997) The first questions looks for a solution of the *inverse problem*, i.e., for reasoning from outcome to action, in order to select an action (Garcia & Morari 1982). For this, the decision maker needs an *inverse model* that determines the action that will bring the system closer to the desired goal state. Question (b) requires a model that predicts the expected effect of an action on a given state of a system. In process control literature, such a model is referred to as a *forward model* (or *action model*) (Gibson et al. 1997).

The inverse model may interact with the forward model to select an action that is expected to have the desired effect. The selected action is then performed, changing the

state of the system, and the forward model computes the expected effect of the move (Gibson et al. 1997). Next, the state of the system is observed, the computed expectation may be compared with the actual outcome to update the forward model, and the cycle repeats. This is illustrated in Figure 1 below.

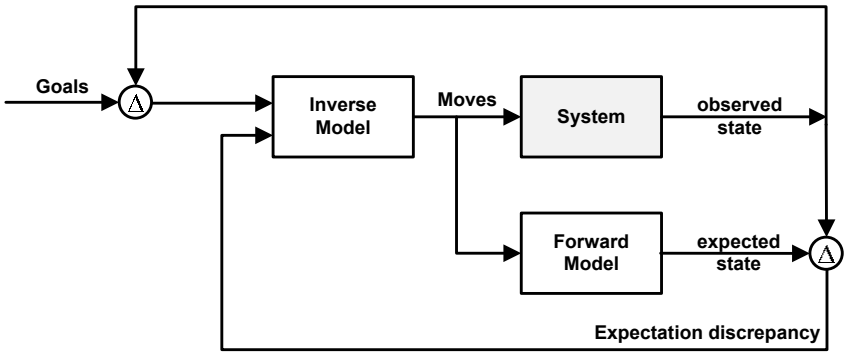

Figure 1. Dynamic decision making as process control

Conant and Ashby (1970) proved that "any good regulator of a system must be a model of that system". This implies that an effective decision strategy must contain a model of the system it aims to control. In the dynamic decision making context, Brehmer (1990) refers to such models as *mental models*. For example, a physician treating a patient must have a mental model of the patient's disease process at an appropriate level of aggregation in order to choose treatments that bring the patient to desired goal states. The disease process itself is hidden, but the physician's mental model contains an expectation that a certain medication will, for example, lower a patient's blood pressure, and the physician selects the medication as a viable treatment option based on this expectation. More generally, a mental model must represent goal-relevant aspects of the

system it controls (J. K. Doyle & Ford 1998). A mental model contains a representation of the state of the system being controlled, the goal states, and expectations of the effect of the actions the decision strategy may choose. In dynamic decision making, the mental model functions as a forward model.

Brehmer (1990) states that there are two major categories of strategies from control theory: feedback and feedforward. A *feedback strategy* makes decisions based on the current state of the system and the effect of previous actions, i.e., the output of the system is fed back as input. The feedback strategy has a relatively simple forward model, because it does not factor predictions into its decision making. In contrast, a *feedforward strategy* attempts to anticipate changes in the system, both as a result of its actions and from outside disturbances, and adjusts its actions accordingly. In order to anticipate changes, feedforward strategies require a more sophisticated mental model than feedback strategies.

Using the process control framework, a successful control strategy brings a system to a goal state. The research proposed here is concerned with how some decision strategies are able to succeed at reaching goal states while others fail. The following section theorizes about the nature of contributions to success and failure.

1.3. Determinants of Decision Strategy Performance

The performance of a decision strategy is determined both by features of the environment as well as by features of the strategy itself.

Available actions. The environment determines what actions are available to a decision strategy. For example, in disease care, constraints imposed by the environment may include the treatments available to treat a certain condition, whether a patient can safely use these treatments (no contraindications), treatments that have already been used, and a patient's ability and willingness to accept or comply with a certain treatment.

A strategy can only succeed to the extent that the available actions can bring a system to a goal state. The need for a sufficient set of actions can be framed in terms of Ashby's Law of Requisite Variety (Ashby 1968), which states that, for a controller to be successful, it must have sufficient variety (range of actions) to match the variety that exists in the system to be controlled. For example, consider Table 1 which represents outcomes of three different treatments (A, B, C) for five patients. ✓ denotes a desirable outcome; × an outcome to be avoided. If only two of these treatments were available, one of the five patients could not avoid an undesirable outcome: patient 1 if C is not available, patient 3 if B is not available, and patient 4 if A is not available. In other words, there exists variety in the patients that requires at least three different treatment options to bring everyone to a desirable outcome.

Table 1. Example of variety in outcomes for different patient/treatment combinations

	Treatment A	Treatment B	Treatment C
Patient 1	×	×	✓
Patient 2	✓	×	✓
Patient 3	×	✓	×
Patient 4	✓	×	×
Patient 5	×	✓	✓

✓ desirable outcome; × undesirable outcome

Delay of feedback. If the effect of actions taken by a strategy is delayed, the challenge of controlling a target system is increased (Sterman 1989). If a delay exists such that the effects of past actions have not (fully) materialized by the time of the next decision, the decision maker is unable to attribute success or failure to prior decisions. Furthermore, if a decision strategy fails to account for the delay, it fails to consider pertinent information and mistakenly assumes that the current state incorporates the results of all prior actions taken.

Observability. Decision strategies make decisions based not on the true state of the world, but on their mental representation of that state. If the true state of relevant variables is hard to assess (for example, it can only be measured through proxy variables or there is a lot of measurement error), there is increased potential for discrepancies between estimates of the state based on the mental model and the true state, which will affect the actions taken.

Stability. In a stable environment, the relationships between actions and their effects do not change. This allows the decision strategy to rely on these relationships when making predictions of the effect of actions taken. If the environment is unstable, predictions will diverge from the actual results obtained by implementing actions. An example of an unstable environment is a patient with varying adherence, who may take medications as prescribed some days but fail to take them on others, with no discernible pattern. The physician will not see the expected treatment effects because the

feedforward prediction is making an assumption about the patient's adherence that is incorrect.

Delay of feedback, measurability, and stability all contribute to the uncertainty present in dynamic environments and pose challenges to a decision making process. In addition, there are features of the decision strategy itself that may limit its ability to succeed.

Inverse model. The inverse model contributes to the performance of a decision strategy by generating and selecting actions for a strategy. If the inverse model fails to generate actions that move a system towards goal or erroneously generates actions that move the system away from goal, the strategy's performance is negatively impacted. These failures can occur because the inverse model misperceives the current state of the system, because, for example, delay of feedback exists, or because the relationship between action and outcome is inaccurately or incompletely modeled. An example of the latter is failure to consider side effects that negatively affect the system's progress towards goal.

Mental model. The mental model contains representations of the state variables as well as expectations for the results of actions taken. If the mental model fails to include relevant variables or account for delay of feedback, the decision strategy is limited in its ability to choose the best available action. Similarly, if the expected effect of actions differs from their true effect, the strategy may choose actions that do not have the desired effect.

Feedback vs. feedforward. Two major categories of process control strategies have been identified: feedback and feedforward (Brehmer 1992). A feedback strategy acts based on the information at hand to reduce the difference between the current and the goal state. Feedback strategies are not anticipatory; they judge their actions based on the observed result of these actions at the next decision point. Feedback strategies are negatively impacted by the presence of delay of feedback. In contrast, feedforward strategies predict the consequences of their actions and take these predictions into account when making choices. A feedforward strategy requires a more sophisticated mental model in order to generate predictions, and is more prone to failure in unstable environments because its predictions may not hold.

To examine the role of these determinants of performance, the research proposed here must be grounded in specific dynamic decision context. The context of diabetes care was chosen for this research. In addition to being a good example of a dynamic environment, it is also a context of interest to practitioners, as discussed in the following section.

1.4. Example Context: Diabetes Care

Diabetes management is an instance of a dynamic decision problem in the healthcare domain. The goal of treatment in diabetes care is to bring a patient's blood sugar, blood pressure, and cholesterol to evidence-based goals in order to prevent *adverse clinical events*, such as heart attacks and strokes, and co-morbidities such as kidney failure and blindness. The treatment requires regular office visits at which physicians monitor

progress and adjust treatment regimens as necessary. Treatments interact in complex ways; for example, a treatment intended for lowering blood pressure can also negatively affect indicators for kidney function and consequently contraindicate a given blood glucose treatment. The disease progresses on its own but is also affected by the physician's treatment actions. Physicians formulate strategies for the treatment of chronic diseases by adopting tools to facilitate the decision making process, for example, by using lab tests, clinical guidelines, formularies, and reference materials (PDR Staff 2011). Lastly, decision makers in healthcare often have to balance competing goals, for example, bringing patients to evidence-based goals, reducing the risk of adverse events, and managing costs. Diabetes treatment involves trading off treatment (outpatient) costs for costs of complications (inpatient). Treatment costs include the cost of visits, lab tests, and medications; complication costs are the costs resulting from the immediate treatment of an adverse event, such as heart attack or kidney failure, as well as follow-up costs for the remainder of the patient's life.

The treatment of diabetes is a problem of high societal interest for a number of reasons. In the United States in 2007, 23.6 million Americans were estimated to be diabetic and another 57 million to be pre-diabetic, i.e., have above-normal levels of blood sugar. Diabetes doubles to quadruples the risk of heart attacks and strokes, increases the likelihood of having high blood pressure, can cause blindness and kidney failure, and sometimes makes it necessary to amputate limbs. In addition, diabetes has a very high cost to society. In 2007, diabetes was responsible for $116 billion dollars of direct

medical costs and another $58 billion in indirect costs such as disability, work loss, and premature mortality (American Diabetes Association 2007).

1.5. Example Decision Problem: Prioritization of Care

Diabetes patients who also suffer from hypertension and/or hyperlipidemia are particularly at risk for adverse events such as heart attacks and strokes. These patients are referred to as *complex patients*. The goal of disease management for patients with one or more chronic diseases is to lower their risk of adverse events. However, treating such patients is challenging because physicians have to choose among conditions to treat. It is often not feasible to treat all conditions at once, for reasons of medical safety, concerns about treatment interactions, side effects, or because of patient choice (Piette & Kerr 2006). Moreover, since the time of a given physician/patient encounter is limited, the different conditions represent competing demands and generally the most pressing or symptomatic patient concerns are addressed while others are deferred to future encounters (Parchman et al. 2007; Jaen et al. 1994). The physician's task of choosing what to treat at a given visit with a patient is referred to as the *prioritization of care* problem. Prioritization of care is the central problem that grounds the research proposed here.

The following patient state variables have been found to be risk factors that significantly correlate with the occurrence of adverse events: age, sex, height, weight, ethnicity, blood glucose level (measurement: A1c), blood pressure (measurement: systolic blood pressure SBP), cholesterol (measurement: low-density lipoprotein LDL),

and history of atrial fibrillation (Stevens et al. 2001). Cardiovascular risk can be separated into *irreversible* and *reversible risk*. Some risk factors, such as age, sex, and height cannot be affected by treatment and therefore contribute to the patient's irreversible risk. Others, such as blood pressure, glucose, cholesterol, and weight can be affected by diet, exercise, and medication, and are therefore reversible risk factors. These are the factors of interest for prioritization. Two approaches have emerged in the literature to guide the treatment of conditions associated with reversible risk factors: treatment to evidence-based goals and risk-based treatment (Institute for Clinical Systems Improvement 2009; D'Agostino et al. 2008; Mazze et al. 2007).

Treatment to evidence-based goals. Evidence-based goals are typically established through committees reviewing evidence from practice and clinical trials (Briss et al. 2000). A threshold for a given patient characteristic, such as blood glucose level or blood pressure, is established such that levels below this threshold are considered "under control", e.g., are associated with significantly lower levels of risk (Briss et al. 2000; Devereaux & Yusuf 2003). For example, a person with a systolic blood pressure greater than 130mmHg is considered hypertensive and at greater risk for a heart attack or stroke. The thresholds established in this fashion then become goals for treatment by physicians. For the purposes of this research, the following evidence-based goals are assumed (Mazze et al. 2005).

Table 2. Evidence-based goals for type 2 diabetes patients

Condition	Measure	Goal
Hypertension	Blood pressure	SBP ≤ 130mmHg
Hyperglycemia	Blood glucose	A1c ≤ 7%
Hyperlipidemia	Cholesterol	LDL ≤ 100mg/dl

Source: *Staged Diabetes Management* (Mazze et al. 2005)

Risk-based treatment. For risk-based treatment, a physician uses an estimate of a patient's risk and attempts to reduce this number through the treatment of risk factors. A benefit of this approach is greater freedom in the selection of treatment options. If a similar risk reduction could be obtained by taking a blood pressure medication or by losing weight, a patient might prefer one option to the other.

The strategies which guide the decision making with respect to prioritization of care are referred to as *prioritization strategies* or *treatment strategies*. The role of their features, such as mental model and feedback vs. feedforward control, will be investigated in this research. In this chapter, the background and context for the proposed research were established. In the following chapter, the research goals and objectives are formulated, both in general terms, as well as in the context of diabetes care.

2. Goals and Objectives

Decision making is a goal-oriented activity. The decision maker is tasked with modifying the state of a system to bring it to a goal state. The goal states are domain-specific. For example, goals of a business may be to increase profits, i.e., increase revenue or save costs. In chronic disease care, the goals may be to prevent complications or lower the cost of treatment. A given system may have only a single state, or many. Some states may be better than others, for example, by being closer to a desired goal state. Therefore, in addition to the strategy itself, a metric to assess the results of the strategy is needed. In many domains, such metrics are defined in terms of outcomes or benefits as well as costs, as described below.

2.1. Evaluation of Decision Strategy Performance

A decision typically results in an action that affects the environment in order to bring it closer to a desired state, referred to as goal state. A *successful* decision strategy eventually brings the system being controlled to a goal state. The ability to bring a system to a goal state is referred to as *efficacy* or *effectiveness* of a strategy. At the same time, actions that are implemented based on decisions incur *costs*, e.g., they consume resources (such as medications, time of providers, organizational resources). In many contexts, there is a trade-off between benefits (a result of efficacy) and costs, referred to as *cost-effectiveness*. For example, in the diabetes care example, the objective may be to lower risk of heart attacks or strokes but to keep the cost of treatment below a given threshold. This is illustrated in Figure 2.

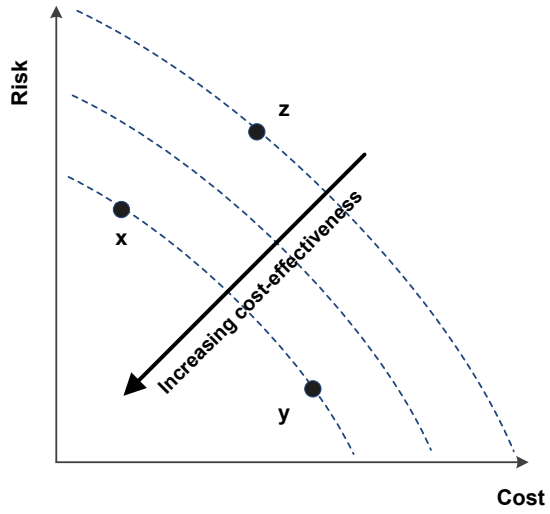

Figure 2. Cost-effectiveness framework for strategy evaluation

The performance of a decision strategy can be operationalized as an objective function $\Phi(s)$ that assesses the quality of a given system state s. For two states x and y, we say x is better than y if, and only if, $\Phi(x) < \Phi(y)$. The decision maker's goal can be stated as minimizing $\Phi(s)$ or lowering $\Phi(s)$ below a goal threshold G. In this example, the dotted lines represent *isobars*, equal values of Φ; in Figure 2 strategies x and y perform equally well in terms of cost-effectiveness ($\Phi(x) = \Phi(y)$), and both x and y perform better than strategy z, expressed as $\Phi(x) < \Phi(z)$ and $\Phi(y) < \Phi(z)$.

In general, cost-effectiveness can be improved in two ways: (a) greater efficacy for the same level of cost, or (b) lower costs for the same level of efficacy. The research proposed here is concerned with questions of what decision strategies must be like to

succeed or fail to achieve specific outcomes in a given context, i.e., what helps or hinders their efficacy or cost-effectiveness?

The performance of decision strategies is of interest on two levels: individual and population. A decision strategy may perform well for a population, on average, but leave specific individuals worse off than another strategy. Consider the example payoff matrix in Table 3. Strategy *a* obtains higher payoffs, on average, than strategy *b*. It does so by producing exceptional payoffs for two individuals but obtaining zero or negative payoffs than *b* for the remainder of the population. Strategy *b*, on the other hand, performs equally well for everyone and all 5 individuals obtain positive payoffs.

Table 3. Example payoff matrix for two strategies

Individuals	Strategy *a*	Strategy *b*
1	20	5
2	-5	5
3	20	5
4	-5	5
5	0	5
Mean	6	5
Proportion of individuals with positive payoffs	.40	1.00

In the clinical context, the focus is on obtaining the best outcome (e.g., highest payoff). Patients (and the provider) will not be concerned that a strategy performs well for the population on average; their focus is on their particular N=1 case. On the other hand, organizations and health systems are interested in the performance of strategies on the population level because they view costs and events in aggregate. It may be worth spending more on an individual patient at high risk for an event because the system has to

absorb much of the cost of the adverse event if it occurs. In order to provide a comprehensive picture of decision strategy performance, both individual and population metrics are considered.

2.2. Research Objectives

Heiner (1983) argues that uncertainty in the environment gives rise to predictable behavior. Uncertainty requires methods that work not only for a particular state but for regions of a state space. Decision strategies represent such methods for decision making problems. The research proposed here is focused on the performance of such decision strategies. In order to understand performance, we first need to understand what aspects or features of decision strategies drive performance. Next, we can conduct experiments in which we modify these features in order to affect performance. Since a strategy's performance is a function of its ability to bring a system to goal, and the goal is domain-dependent, this investigation must take place in a particular problem context. For this research, the prioritization of care problem for diabetes patients has been selected as the context. Below, the research objectives are described and related to the diabetes care context. The specific performance measures for diabetes care are discussed in Section 2.3.

Objective 1. *Determine the impact of decision strategy features on performance*.

To determine the impact of strategy features on performance, a set of strategies that differ in these features will be created and evaluated. Three aspects of the strategy will be

examined: feedback vs. feedforward control, the amount and type of information available to the strategy, and the complexity of the computation in the inverse model.

Feedback vs. feedforward control will be varied by changing what expectations for its actions are available to the strategy in its forward model. Feedback control only requires sufficient information to relate an action to its goal, for example, by having an expectation that a blood pressure medication will lower blood pressure. A feedforward strategy requires a more specific prediction, e.g., an expectation of how much a given blood pressure medication will lower blood pressure. For a simpler forward model for a feedforward strategy, this expectation may be the same for any blood pressure medication; a more complex strategy may contain different expectations for different medication classes and doses.

The amount and type of information available to the mental model determine what can enter the computations of the inverse and forward models. For example, if blood pressure information is not available, it cannot be used to decide on a blood pressure move or to make predictions about how it can be affected by treatment. If a strategy has reference information, such as the distribution of a given state variable in the population, it can assess the relative standing of a given instance to the population. For example, if a strategy has a mental representation of the blood pressure distribution in its population, it can determine a given patient's percentile, and subsequently could determine if it is dealing with a usual or extraordinary case and adjust its actions accordingly.

The complexity of the computation in the inverse model determines which choices a strategy can make. A simple strategy may compare a state variable to its goal and act if the goal state has not yet been reached. This strategy only has binary information about being at goal and can therefore only decide whether to make a move or not with respect to this state variable. A more complex strategy may compute a distance-to-goal and choose different moves based on the size of this distance. For example, if a distance-to-goal metric is used to group patients into low, medium, and high blood glucose, the strategy can employ up to three different moves for the different categories.

To determine the impact of strategy features on performance in the diabetes care context, a constrained version of the prioritization of care problem will be constructed. The problem will be constrained such that it becomes possible to search the space of possible treatment actions exhaustively. Using exhaustive search, we can determine the best possible performance (using the measures described in Section 2.3 below) that a strategy could achieve in this idealized formulation of the problem. This makes it possible to compare the performance of strategies to an optimum and potentially identify strategies that achieve optimal performance. The best-performing strategies (i.e., the ones achieving optimal or closest to optimal performance) will be analyzed to determine how their features interact with the environment to achieve this level of performance.

One reason why a strategy may fail to achieve optimal performance is because of limitations in terms of its mental model or inverse model. Another reason is that a strategy may commit *errors*, i.e., moves other than the ones resulting in good

performance. If this is the case, a sub-optimal strategy can be improved by reducing or eliminating errors. This is the focus of Objective 2.

Objective 2. *Determine means for modifying existing decision strategies to compensate for errors and, consequently, improve performance.*

A strategy may fail to achieve desirable performance because it fails to take sufficient action (*error of omission*) or acts too strongly (*error of commission*) (Ramsey 2010). In a cost-benefit framework, an error of omission results in a failure to obtain benefits by failing to take action, but has a lower cost. Errors of commission can be categorized into two groups: harmful action and wasteful action. An action is harmful if it decreases benefits (for example, over-treating a patient in such a way that it results in complications); it is wasteful if it does not affect or increases benefits but at a greater cost than necessary.

If errors occur and can be reliably identified or predicted, and strategies can be modified to compensate for these errors, strategy performance can be expected to increase. For Objective 2, a systematic approach will be developed to predict errors and take compensating action. In the diabetes care context, this approach will be used to increase the performance of existing treatment strategies in terms of their efficacy and cost-effectiveness. As in Objective 1, the strategies resulting from this approach will be analyzed to determine how their features interact with the environment to achieve their performance.

While Objective 2 is focused on examining and compensating strategies for specific failures that result from the decision strategy, Objective 3 investigates success and failure as a function of the environment. In addition to being successful due to strong mental models and appropriate control, decision strategies may be successful because they are well-adapted to a particular problem environment. Gigerenzer refers to such strategies as *ecologically rational* (Gigerenzer & Todd 1999; Gigerenzer 2001). The power of the environment is illustrated by the ability of zero-intelligence traders that purchase and sell randomly in a market with minimal constraints to be highly efficient (Gode & Sunder 1993). Since the traders possess zero intelligence, this performance cannot be a result of their mental model or mode of control; instead, it must be a result of their environment. Consequently, the type of decision strategies that can be evolved in a given environment is of interest, which is Objective 3 of the proposed research.

Objective 3. *Determine means of evolving decision strategies for a given problem environment.*

Objective 2 focused on an approach to improve existing decision strategies by reducing or eliminating errors of omission or commission. In contrast, Objective 3 focuses on the development of an approach to evolve decision strategies to achieve a desired level of performance. Do strategies which evolve from such an approach mimic existing strategies or do they use information available in the environment in new and interesting ways? Do they obtain outcomes in terms of efficacy or cost-effectiveness that are not reached by existing strategies or strategies improved from existing strategies in

the investigation of Objective 2? These questions are the focus of the third and final objective.

2.3. Performance Measures for Diabetes Care

2.3.1. Efficacy

Diabetes and its two most commonly associated co-morbidities, hypertension (high blood pressure) and hyperlipidemia (high cholesterol), are medical conditions that considerably increase the likelihood of complications such as heart attack, stroke, blindness, kidney failure and amputation. The primary objective of clinical care in this context is the prevention of these complications by controlling diabetes, blood pressure, and cholesterol.

For the research proposed here, the efficacy metric for the population level is an expected *cardiovascular event rate*, i.e., the proportion of a specific population that experiences a heart attack or stroke in a ten year follow-up period. More efficacious strategies will result in lower event rates or, equivalently, in a greater number of prevented events.

On the individual level, efficacy is measured in terms of ten-year risk. *Risk* is defined as a patient's probability of experiencing an adverse clinical event, such as a heart attack or stroke, in a specified period of time. Risk is a function of *risk factors*, patient variables that have been shown empirically to be associated with higher or lower likelihoods of an event occurring. For example, for diabetes patients, the following risk factors are typically considered: age, gender, ethnicity, height, history of atrial fibrillation, smoking,

blood glucose level (A1c), systolic blood pressure, cholesterol, and weight (D'Agostino et al. 2008; Stevens et al. 2001; Clarke et al. 2004). Risk factors are either *reversible* or *irreversible*. A risk factor is irreversible if it cannot be changed or can only change in one direction. Ethnicity is an example of the former, age is an example of the latter. Reversible risk factors can be changed in both directions, and lowering these risk factors (in order to reduce risk) is a goal of risk-based treatments such as diet, lifestyle, exercise, and medication. It is important to note that, while a factor may be reversible, the associated risk may not be fully reversed. If a patient who has smoked for 20 years gives up smoking, their smoking status will be reversed, but this does not necessarily lower their risk to the same level as that of a person who has never smoked. In other words, risk is path-dependent. Furthermore, not all reversible risk factors represent treatment options for every patient. For example, a patient may choose not to give up smoking regardless of the risk benefits.

2.3.2. Costs

There are two major categories of costs in chronic disease care: inpatient and outpatient. *Outpatient costs* are incurred as a part of treatment; they are the expenses related to visits, lab tests, and prescribed medications. Inpatient costs are the costs associated with complications, such as heart attacks and kidney failure. They require hospitalization (or other clinical care, such as emergency room or urgent care visits) and intensive treatment at the onset of the adverse event. In addition, once a patient has experienced an adverse event, there is an annual follow-up cost for the remainder of the patient's life (Clarke et al. 2004).

To evaluate strategy performance, the cost of treatment can by using a lookup table that gives costs for actions chosen by a decision strategy such as visits, lab tests, and medications. Strategies can be modified to manipulate these costs, for example, by increasing or decreasing the frequency of visits.

The cost of complications, on the other hand, is usually an estimated cost, calculated as the expected value given a patient's risk for a particular event. The costs for the initial hospitalization and annual follow-up for complications used in these studies are taken from Gilmer et al. (2012). These costs are affected by treatment; however, they cannot be directly controlled.

2.3.3. Cost-Effectiveness

A decision strategy in diabetes care selects treatment actions and, consequently, incurs treatment costs (outpatient costs). Cost-effectiveness is determined by looking at what is gained in efficacy compared to additional outpatient spending. Given that efficacy is measured in terms of risk (for individuals) or event rates (for the population), the cost-effectiveness metrics are ratios, namely the *cost per 1% risk reduction* and *cost per prevented event* ratio. If two strategies can prevent the same number of events, the less expensive strategy is said to be more cost-effective. In Figure 3, strategies with equal cost-effectiveness are represented as residing on the same *isobar* (dashed line).

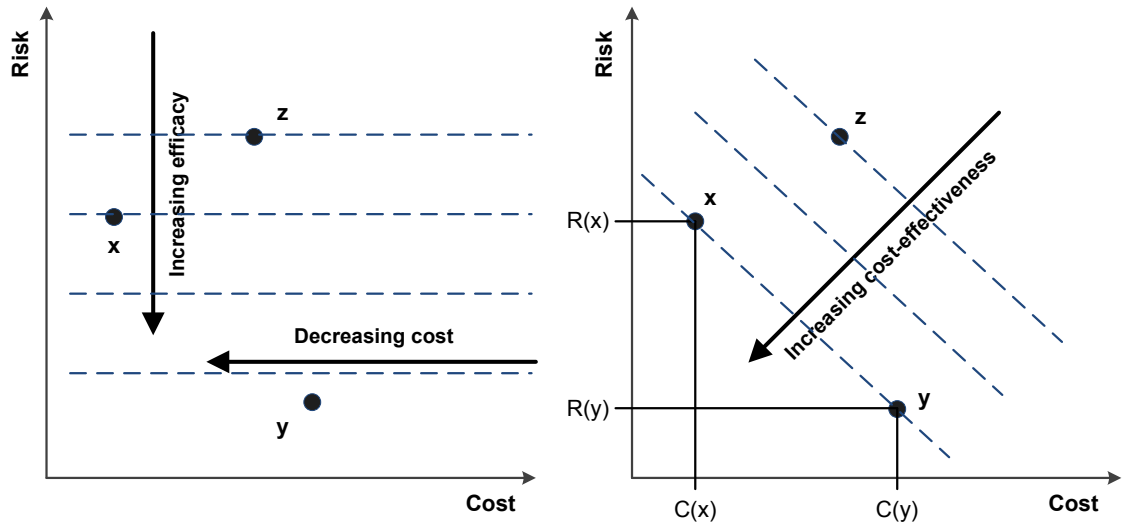

Figure 3. Efficacy (left) vs. cost-effectiveness (right)

Cost-effectiveness isobars are defined by the sensitivity to cost. For example, would a patient spend an additional $10 per month to lower cardiovascular risk by 1%? What about $50? Or $100? Let m represent the maximal amount a patient is willing to spend to reduce risk by 1%. Let $C(s)$ be a function that gives the cost for strategy s, and $R(x)$ a function that gives the risk for s. Then two strategies x and y are on the same isobar if, and only if, $C(y) - C(x) = m(R(x) - R(y))$.

The cost-effectiveness question is complicated by the fact that patient spending is typically capped through insurance. A patient on a high-deductible plan may be more cost-sensitive than one on a low-deductible plan. In the idealized setting of this research, patients are assumed to have equal cost-sensitivity, i.e., the isobars are identical for all patients.

When the only concern for a strategy is efficacy, this can be represented as a special case of the cost-effectiveness problem where $m = \infty$, i.e., patients are perfectly insensitive to cost. The isobars in Figure 3 represent this case. The other extreme, in which the focus is cost-reduction regardless of losses in efficacy, would be represented as $m = 0$. The isobars for $m = 0$ would be parallel to the y axis in Figure 3.

The scope of the research proposed here is limited to cardiovascular events. The methodology applied to conduct this research scales to incorporate additional types of events and treatments. To measure the cost-effectiveness when considering multiple types of events, a common approach in health economics is to determine a quality of life value for each type of event (Sassi 2006). This value represents the utility of a year of life after experiencing a complication. These utilities are then used to compute a patient's quality of life adjusted years (QALYs). 1 QALY represents the utility of living for a year with perfect health. For example, the UK Prospective Diabetes Study (UKPDS) determined that diabetes reduces a patient's quality of life such that for a year of life, the patient accrues .785 QALYs (Clarke et al. 2004). Instead of comparing the cost per prevented event, cost-effectiveness in this case is measured by cost per additional QALY.

Table 4 summarizes the performance metrics used in this research for efficacy, cost, and cost-effectiveness on the individual and population level.

Table 4. Performance metrics for individual and population level

Level of Analysis	Efficacy	Cost	Cost-Effectiveness
Individual	Cardiovascular risk	Cost of treatment, estimated cost of complications	Additional cost of treatment per 1% risk reduction
Population	Cardiovascular event rate	Total cost of treatment, estimated total cost of complications	Additional cost of treatment per prevented event

3. Methodology

To accomplish the objectives outlined in the previous section, an appropriate experimental setting is needed. Typically, demonstrating the superior efficacy or cost-effectiveness of a method or treatment in a clinical context requires randomized clinical trials and cost-effectiveness analyses. Because the focus of this research is on the relationship between strategy features and performance, the investigated decision strategies are idealized to emphasize the role of the relevant features; thus, for the research proposed here, cost-effectiveness studies are simulated. Simulation allows for the estimation of within-patient differences of various treatment strategies and enables the creation of a large number of treatment cases in a short amount of time. Additionally, in this context, simulation makes it possible to investigate baseline treatment policies such as "no additional treatment" that could raise ethical concerns in real trials. To simulate clinical trials, the following are needed: a model of type 2 diabetes mellitus (T2DM) patients to create synthetic patients, a population based on which synthetic patients can be modeled as participants in the trial, and a source of the decision strategies under investigation.

The model of T2DM patients must be able to represent the relevant patient state variables and treatment pathways to a sufficient extent that the findings are clinically plausible. The Simcare patient model is such a model; it has been validated by replication of randomized clinical trial studies as well as N=1 case studies. It is discussed in detail in Section 3.2 below. The population on which the synthetic patients are based is a real

population under care at a regional health system. Relevant characteristics from this population are used to initialize a synthetic patient population, while explicit assumptions about other aspects, such as patient adherence and depression, are made. The characteristics of this population are described in Section 3.3. To represent and examine the strategies of interest, a *computational model* of decision strategies will be developed. A computational model considers aspects of a decision strategy guiding treatment as a system that takes inputs and transforms them, by means of a function, into outputs (Newell 1990). This transformation of inputs into outputs occurs by means of computation based on a representation of the system, i.e., based on a model (Marr 1982).

The overall methodology is illustrated in Figure 4; the focus of the research proposed here is the shaded box.

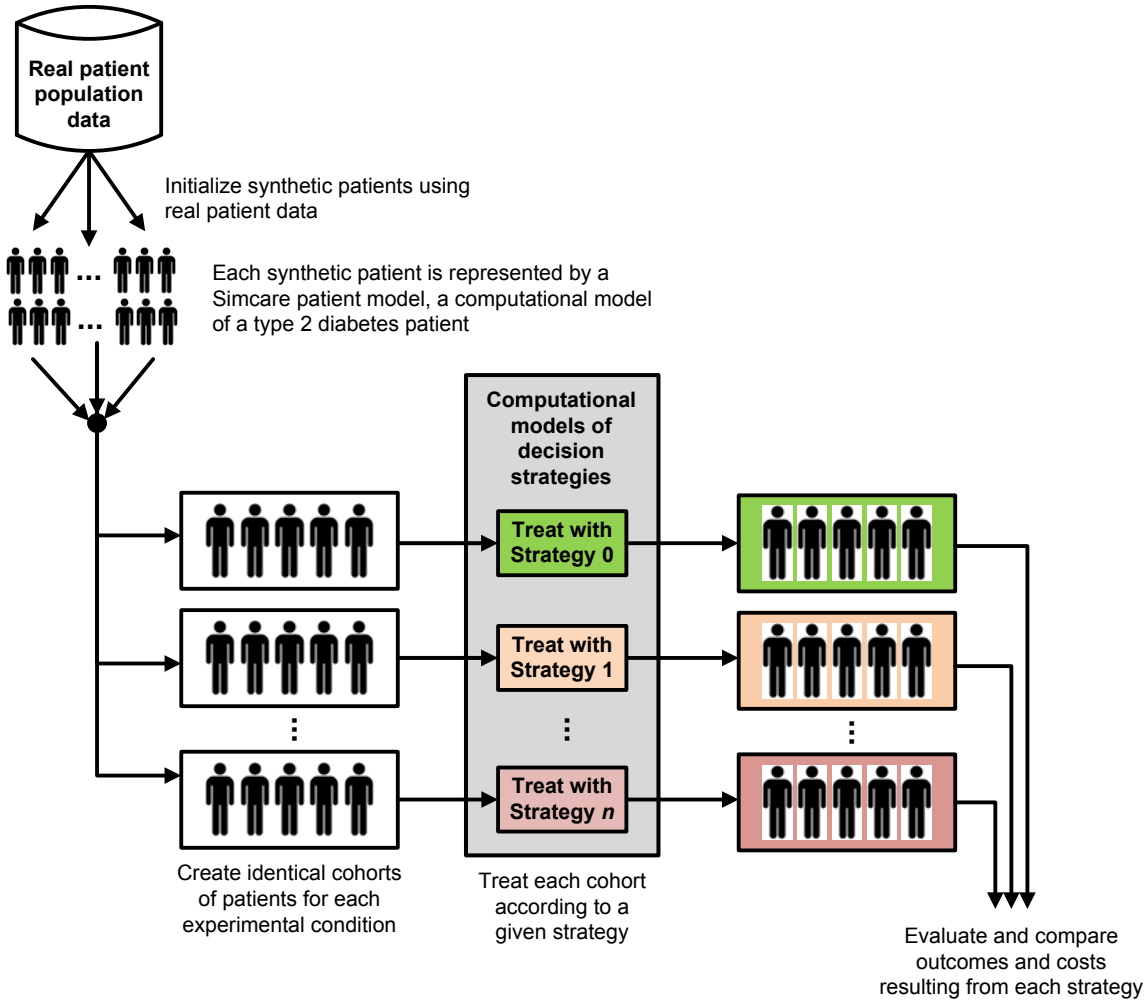

Figure 4. Illustration of overall methodology

3.1. Constraints

Decision strategies select among available actions in order to reach identified goals. A feature of dynamic environments is that available actions are constrained by the current state of the system and previously implemented decisions, as well as the organizational context. The actions available to the physician (and, therefore, the decision strategies) are

constrained in several ways. The choice of drugs is limited by the medications available to the physician. Only a limited number of drug classes are available for each condition, and each drug can only be described in specific increments. Finally, not all drugs are suitable for each patient.

Formulary. A formulary describes which drugs and drug classes are available in a given context to treat a patient. For example, an endocrinologist who specializes in diabetes care will have more drug options than a family practice physician.

Sequence of doses. The first drug of choice for the treatment of blood glucose in type 2 diabetes is typically metformin, an insulin sensitizer. This drug can be prescribed in 500mg, 1000mg, 1500mg, and 2000mg doses. Because of safety concerns and potential side effects, the physician must start at a lower dose and gradually increase (titrate) (Mazze et al. 2007). The same is true for other classes of drugs to treat blood glucose, as well as for the drugs used to treat blood pressure and lipids.

Dose responses. There is generally a diminishing return for higher doses of each drug class (illustrated in Figure 5), i.e., the first dose typically results in a bigger improvement for the associated condition than do following doses (McCabe et al. 2010). This increases the complexity for the decision maker because it may in some cases be preferable to start a patient on a new drug class rather than increment the dose for an existing one.

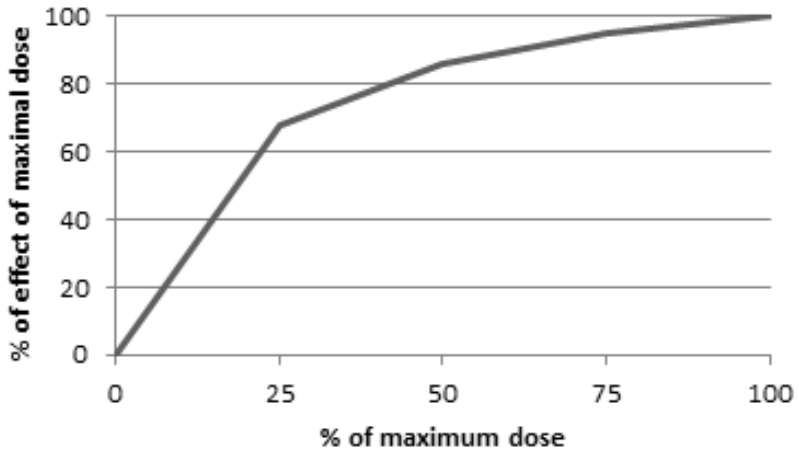

Figure 5. Example dose response curve illustrating diminishing returns for higher doses

Contraindications. Medical guidelines identify conditions when certain drugs should not be prescribed out of concern for patient safety. For example, metformin is contraindicated for patients with serum creatinine greater than 1.5mg/dl. Glipizide, another type of drug to treat blood glucose, must be discontinued if a patient is started on insulin. These contraindications provide additional constraints on the available treatments for some patients.

Delay of feedback. The effect of treatment is not instantaneous; depending on the drug, it can take between 7 and 90 days for a treatment to take full effect (see Figure 6) (Dutta et al. 2005). The delay of feedback hampers a strategy using feedback control because it must either wait until the delay expires, or risk over-treating if it makes a move based on information that does not represent the fully realized drug effect. For this reason, some clinical guidelines impose waiting periods before the next move for a certain condition can be made after giving a particular drug (Mazze et al. 2005).

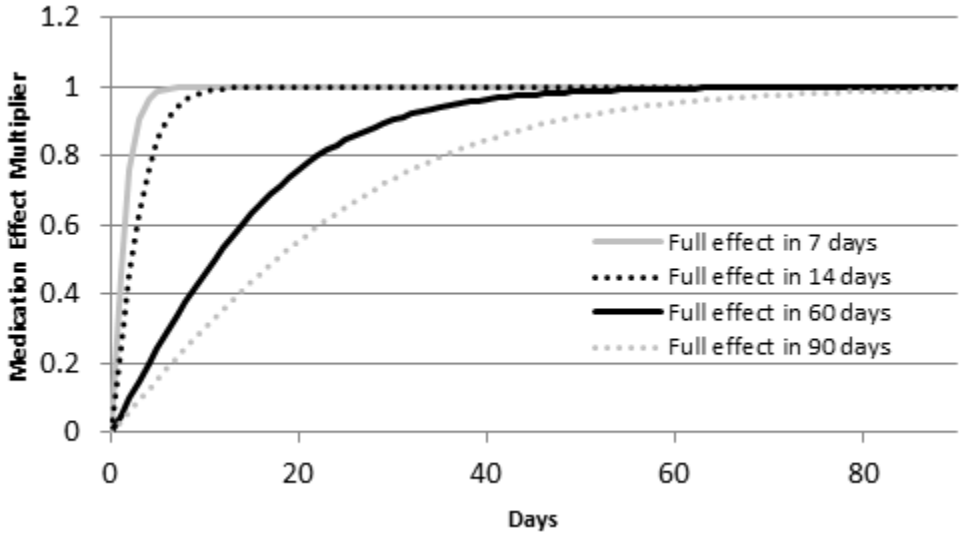

Figure 6. Time effect curves

3.2. Patient Simulation: Simcare Patient Model

To create synthetic patients for the proposed research, the Simcare patient model described by McCabe (2012) is used. This model represents individual diabetes patients and their reactions to clinical treatments. Simcare models patients' blood glucose, blood pressure, cholesterol (lipds), serum creatinine, responsiveness and adherence to medications, and depression. In Simcare, these variables are affected through pathways for the effects of different diet prescriptions (e.g., low sodium, high fiber); exercise; four classes of glucose medications (biguanides, sulfonylureas, thiazolidinediones, insulins); six classes of blood pressure medications (ACE-inhibitors, diuretics, calcium channel blockers, β-blockers, angiotensins, and α-blockers); two classes of cholesterol-lowering medications (statins and fibrates); one antidepressant; and referrals to nurse educators. In addition, Simcare contains pathways for disease progression that reflects deterioration of

a condition over time. For each treatment, the Simcare model contains a dose response curve and a time effects curve, making it possible to observe delay of feedback effects. For example, all blood glucose treatments have a delayed impact on hemoglobin A1c, a frequently used lab test to evaluate a patient's blood glucose.

To be of value, the simulation must be based on a plausible model of diabetes patients. The Simcare model has been validated for coherence and correspondence against medical literature, randomized clinical trials, and observed treatment effects from individual patients in real clinical practice (McCabe 2012; McCabe et al. 2010). The model has been used successfully to teach physicians in residence (Sperl-Hillen et al. 2010). It has also been applied to investigate clinical decision making in real settings (Ramsey 2010).

3.3. Population

The population used to initialize the synthetic patients is a real population of 18,356 type 2 diabetes mellitus (T2DM) patients under care at a regional health system. Descriptive statistics for the risk factors under consideration are given in Table 5; these match exactly between the synthetic and real population. In addition, synthetic patients are initialized with the same medication prescriptions as their real patient counterparts.

Table 5. Population characteristics (N=18,356 patients)

Characteristic	Mean	SD
Irreversible risk factors		
Age	56.8	15.0
Female (%)	48.9	
Reversible risk factors		
BMI (kg/m^2)	33.8	8.6
Hemoglobin A1c (%)	9.1	1.9
Systolic blood pressure (mmHg)	138.3	18.3
Low-density lipoprotein LDL (mg/dl)	141.2	33.1
High-density lipoprotein HDL (mg/dl)	42.4	12.5
Triglycerides (mg/dl)	191.0	116.6
Current smoker (%)	5	

For characteristics that are not directly observable or available in the data set containing the real patient information, the following assumptions were made.

Adherence. Strategies assume, and unless otherwise stated, patients are simulated as having perfect adherence, i.e., they follow their prescribed treatment regimens.

Depression. Depression has a negative impact on adherence. Consistent with the perfect adherence assumption, it is assumed that patients are either free from depression or already treated for this condition.

Ethnicity. The dataset does not contain the ethnicity of patients, which is considered a risk factor. Therefore, an ethnicity value (Caucasian, African-American, or Asian/Pacific) was imputed based on the other factors, so that the proportions of ethnicities from the real dataset were matched.

3.4. Treatment Protocols and Guidelines

The treatments available to decision strategies are defined by clinical guidelines, such as those provided by the American Diabetes Association, the Institute for Clinical Systems Improvement, and the International Diabetes Center (American Diabetes Association 2012; Riethof et al. 2012; Mazze et al. 2012). Of particular interest for this research are the *Staged Diabetes Management* (SDM) guidelines because they provide specific algorithms for sequences of treatments (for example, to treat blood glucose, one can increase treatment from diet and exercise, to a single oral medication, to combination of oral medications, to insulin) (Mazze et al. 2012; Mazze et al. 2007; Mazze et al. 2005). Moreover, SDM provides treatment algorithms for high blood pressure and high cholesterol that occur concurrently with diabetes.

Staged Diabetes Management (SDM) is used to guide the implementation of treatment decisions made by a decision strategy (Mazze et al. 2007). For example, if a strategy chooses to make a single blood pressure move, the SDM guideline is used to determine what that move should be, based on what medications a patient is currently being treated with and what doses are used. SDM provides contraindications and specific treatment rules for medical safety. For example, if a patient is on the maximum dose of a sulfonylurea (a class of blood glucose lowering medication), this dosage should be lowered before starting the patient on insulin. By incorporating these guidelines, the simulated decision strategies apply plausible treatments and follow established treatment principles.

3.5. Cost Estimation

In the diabetes care context, two types of costs must be estimated: cost of treatment and cost of complications. The cost of treatment is the direct result of the actions taken by treatment strategies, and is therefore under the control of the strategy. The cost of complications is the indirect cost resulting from patients experiencing adverse events, such as heart attacks and strokes.

3.5.1. Cost of Treatment (Outpatient)

The cost of treatment consists of the following types of costs: visit cost, lab cost, referral cost, and cost of medications (Gilmer et al. 2005; Gilmer et al. 2006; Gilmer et al. 2012). These costs are assumed to be identical for all patients. Differences in insurance coverage or discount plans are not modeled for the research proposed here.

Cost of visits. The cost of a visit is incurred every time a patient meets with a physician, regardless of whether another information-seeking or treatment move is made at that visit.

Lab costs. For a physician to determine a patient's blood glucose (A1c) or lipids (LDL, HDL, and triglycerides) levels, a lab test is required. A fixed cost is assumed for each type of test.

Referral cost. If a referral occurs, the patient meets with another provider, such as a nurse educator, a psychologist, or another specialist. In the research proposed here, referrals are modeled as having the same cost as a regular office visit.

Cost of medications. The cost of medications is calculated from a proprietary cost table, provided by a regional health system, that assigns a specific dollar amount to every dose of every medication in the formulary.

3.5.2. Cost of Complications (Inpatient)

The cost of complications results from patients experiencing adverse events. This cost is calculated from the UKPDS (UK Prospective Diabetes Study) Outcomes Model, an established model for cost-effectiveness research in diabetes care (Clarke et al. 2004; Gilmer et al. 2012). The UKPDS Outcomes Model calculates the probability of experiencing adverse events, specifically myocardial infarction, stroke, kidney failure, amputation, blindness, and ischemic heart disease, and assigns a cost to each event. In addition, it computes an annual follow-up cost once an event was experienced.

Because the Outcomes Model stems from the United Kingdom Prospective Diabetes Study, its cost estimates have been adjusted to reflect the U.S. cost structure and for inflation (Gilmer et al. 2012; O'Brien et al. 2003; O'Brien et al. 1998).

3.6. Risk Estimation

For the research proposed here, cardiovascular risk, i.e., the risk of heart attack and stroke, is estimated using published risk models based on survival analysis. There are two major risk prediction models that currently serve as the "gold standard" of cardiovascular risk prediction in the chronic disease context: Framingham and UKPDS (D'Agostino et al. 2008; Clarke et al. 2004; Stevens et al. 2001).

Framingham. The Framingham risk equations were developed based on the Framingham Heart Study, a cohort study conducted in Framingham, MA to investigate causes of heart disease and stroke. The study originally commenced in 1948 and now includes two subsequent generations. Based on the study, risk estimation equations were developed that include the following risk factors: age, sex, body mass index (BMI), systolic blood pressure, cholesterol, smoking status, presence of diabetes, and presence of blood pressure treatment (D'Agostino et al. 2008).

UKPDS. The UKPDS risk equations were developed based on the UK Prospective Diabetes Study, a study with a 10-year follow up for diabetes treatment. This risk model was designed specifically for diabetic patients after it was found that the Framingham equations underestimate risk for diabetic patients (Stevens et al. 2001). The following risk factors are considered: age, duration of diabetes, sex, history of atrial fibrillation, ethnicity, smoking status, A1c, systolic blood pressure, and cholesterol levels (Stevens et al. 2001).

Since the focus of this research is on diabetes patients, the UKPDS risk model was selected for risk estimation throughout the studies described below.

3.7. Feedback vs. Feedforward Control

We use the terms *feedback strategy* and *feedforward strategy* as shorthand for strategy employing feedback resp. feedforward control. Feedback and feedforward strategies differ in their use of information and their assumptions about certain aspects of the problem space, especially in the way they represent the patient state and cope with

delay of feedback as well as adherence. These differences are modeled as follows (see also Brehmer 1992; Ramsey 2010).

Patient state representation. Feedback strategies use the patient as a representation of the patient state. Instead of maintaining a complex representation of patient state in the mental model, a feedback strategy employs a simple forward model that contains an expectation that a given treatment will lower a particular variable, for example, "metformin will lower blood glucose". It does not attempt to predict the dose necessary to bring blood glucose to goal. This simple forward model is sufficient for the inverse model to select metformin as a treatment option for blood glucose. Consequently, after making a glucose move, the feedback strategy has no specific expectation for the blood glucose value at the next visit, other than that it should be lower. Instead, it checks the patient's blood glucose at the next visit to determine the effect. In this sense, its mental model uses the patient's body as representation of itself. Feedforward strategies, on the other hand, create their own mental representation of the patient state. This representation is manipulated to predict the effects of different treatment moves.

Delay of feedback. In the research proposed here, there are two types of feedback strategies that differ in how they manage delay of feedback. The *weak* version assumes that treatment effects are fully realized only when there are no further decreases in the treated value; no other move is made if the treated value decreased between the previous and current visit. The *strong* version operates under the assumption that all medications reach full effect after 90 days and will take another action after this amount of time.

Feedforward strategies, in contrast, anticipate delay of feedback by using a mental representation of the dose response and time effects curves (e.g., in terms of single points on these curves).

Adherence. Feedback and feedforward strategies start with an initial treatment move (visit with nurse educator) to address adherence and then assume a constant, perfect level of patient adherence. Feedback strategies continue with this assumption; feedforward strategies, however, can detect a lack of adherence because of discrepancies between the observed effect from treatment and the effect predicted by the forward model.

4. Study 1: What Determines the Performance of Decision Strategies in Uncertain Environments Requiring Prioritized Action?

A key objective of diabetes management along with the treatment of hypertension and hyperlipidemia is the prevention of cardiovascular and other complications. Complications reduce patients' quality of life and are costly both for patients and health organizations. Effective treatment reduces the risk of complications. However, as discussed above, there are constraints that prevent physicians from making all possible risk-reducing treatment moves at a given visit. Therefore, prioritization of care becomes important. Ideally, prioritization results in treatments that minimize risk as far as possible given the constraints and, therefore, avoids expenditures on treatments that do not significantly impact risk. The goal of this study is to provide a comprehensive comparison of decision strategies used for prioritization of care, both in terms of their efficacy and cost-effectiveness.

In this study, a constrained version of the prioritization of care problem is formulated to analyze strategies observed in practice as well as proposed new strategies. In this version of the problem, the question of prioritization is what condition should be treated at a given visit. The physician has to choose between up to four options: treat blood pressure, treat glucose, treat lipids, or treat nothing. Once a condition is selected, the decision strategy follows a standard algorithm to select the specific treatment move for

that condition (Mazze et al. 2005). Each treatment option is only available if the patient is above the evidence-based goal for the associated condition (see Table 6).

Table 6. Example treatment actions in prioritization of care

Action	Action Description	Action available if
B	treat blood pressure	SBP > 130mmHg
G	treat glucose	A1c > 7%
L	treat lipids	LDL > 100mg/dl
N	do nothing	

Prioritization is most important when treatment opportunities (visits) are limited. In usual care, patients have been observed having between 2 and 4 visits per year during which a chronic condition (e.g., diabetes, high blood pressure, or high cholesterol) is treated (Wagner et al. 2001). To reflect these conditions, the prioritization of care problem is strongly constrained for this study in the following ways. The number of visits is limited to four, visit intervals are fixed at 90 days, and the amount of treatment choices is limited to a single move on a single condition at each encounter. (These constraints represent the base case; sensitivity analyses will be conducted in which these values are changed to determine robustness on findings.) The objective of a prioritization strategy then becomes to select the best sequence of for moves for a given patient, i.e., to make the four moves that lower risk the most. The described constraints make it possible to construct a "game tree" of all possible treatment choices for a constrained version of the problem (illustrated in Figure 7). Using this game tree, it is possible to determine the best outcome in terms of risk reduction for a given patient that can be obtained by treatment given the constraints.

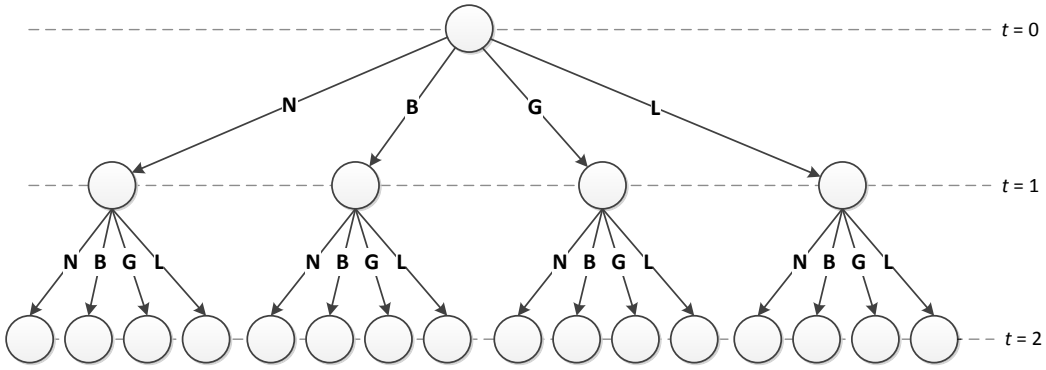

Figure 7. Game tree of possible treatment choices (N = do nothing, B = treat blood pressure, G = treat blood glucose, L = treat lipids)

For the purposes of this study, a *prioritization strategy* selects one condition to treat from all the conditions for which a patient is above goal at a given visit. We consider three of the most common conditions associated with T2DM: hyperglycemia, hypertension, and hyperlipidemia (Mazze et al. 2007). Once the condition is selected, the physician makes a treatment *move*, i.e., takes an action to bring this condition closer to goal.

Each move represents choosing one of the arrows from a node in the tree. A decision strategy results in the selection of particular path through this tree for a given patient. For each path through the game tree, the resulting risk as well as costs can be computed (as discussed in Section 3.4) to determine the efficacy, cost, and cost-effectiveness of the different prioritization strategies. The following strategies are investigated.

Basic Strategies

Random choice. This strategy serves as a baseline to determine the benefit of different prioritization strategies. It randomly selects one of the conditions above goal at each visit and applies a corresponding treatment move. This is not intended to serve as a real decision strategy but rather as a point of comparison for the performance of more principled decision strategies.

Serial treatment. Serial treatment brings one condition to goal (using a sequence of moves), then the next, until all conditions are at goal. For example, a strategy called "BGL" will work on blood pressure until it is at goal, then work on glucose, and once blood pressure and glucose are at goal, will start treating lipids. In the modeled example, there are six possible serial strategies: BGL, BLG, GBL, GLB, LBG, LGB. Serial strategies do not change what condition they treat until goal is reached for that condition. An example of the path chosen by a serial LBG strategy is shown in Figure 8. Serial treatment is a strategy using feedback control with a simple mental model, its only computation is to checks whether a condition is above goal, if so, that condition gets treated, otherwise the strategy checks the next condition in its preset sequence.

Distance-to-Goal Strategies

A distance-to-goal strategy calculates a distance-to-goal metric for each condition at each visit and selects the one with either the greatest (farthest-from-goal) or smallest (closest-to-goal) distance for treatment at that visit. The choice of distance metric determines what type of mental model and decision strategy is required. We consider the

following metrics with increasing requirements on the mental model: percentage, percentile, and expected number of moves to goal. In addition, a risk-based strategy will be shown below that uses reversible risk as a distance-to-goal metric. The first three metrics stem from *treatment to evidence-based goals* (as discussed in Section 1.5), the latter was derived from the *risk-based treatment* paradigm.

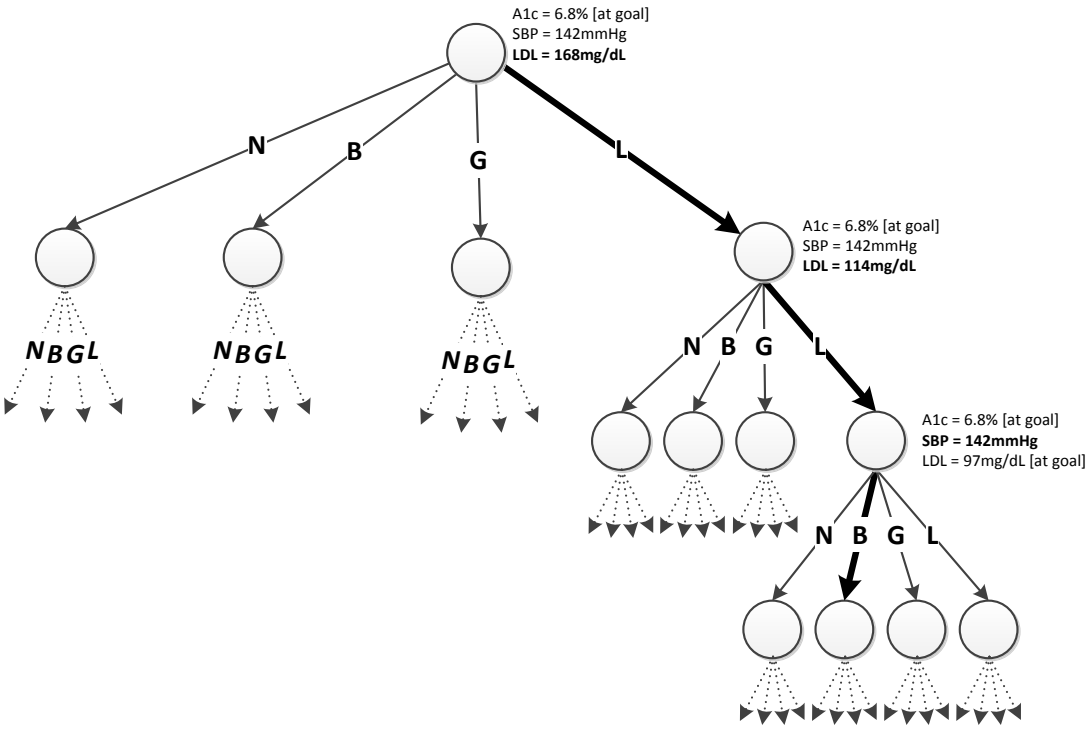

Figure 8. Example of a treatment path resulting from a serial LBG strategy

Metric: Percentage. Percentage is a naïve and easy to calculate distance-to-goal metric that simply takes the percentage by which a given value (for example, systolic blood pressure) exceeds the evidence-based goals. For example, if a patient has an SBP

of 140mmHg and the goal is 130mmHg, the patient is 140/130 = 7.7% above goal for this condition. This metric has minimal requirements for the mental model and can be used by strategies that use feedback control because it can be determined by the information at hand without using expectations.

Metric: Percentile. Percentile considers how far a patient is from goal for a given condition in comparison with the rest of the patient population, expressed as the patient's percentile. The use of this metric implies that the mental model contains a distribution of values to be treated (for example, A1c, SBP, and LDL). For example, if a patient has an A1c of 8% and SBP of 160mmHg, and A1c of 8% is the 80th percentile in the population and SBP of 160mmHg is in the 95th percentile, the farthest-from-goal metric would select blood pressure for treatment because the percentile is greater. It is suitable for feedback control because no forward-looking computation is required.

Metric: Expected number of moves. Using expectations of the effect of treatment moves, the mental model can compute the expected number of moves that must be made to get to goal. For example, if the expectation is that a blood pressure move will reduce SBP by 5 points, to bring a patient from the current 145 to the goal of 130mmHg, the expected number of moves would be three. This metric can only be used by a feedforward strategy because its decision making is based on expectations for treatment moves that must be a part of the mental model.

These metrics can be considered as different ways of ranking patients for treatment. For example, a patient with a SBP of 220mmHg may be ranked considerably higher than

a patient with SBP 190mmHg in terms of a percentage metric (69% vs. 46%); however, they are both in the 99th percentile so they could both be perceived as extraordinarily high blood pressure patients. The expected number of moves takes into account expected treatment effects and would rank the first patient considerably harder to treat (bring to goal) than the second. The 30mmHg difference would likely require six additional treatment moves to bring to the first patient to goal.

Given a metric, a distance-to-goal strategy can prioritize either by starting with the greatest distance to goal or with the smallest.

Farthest-from-goal. A farthest-from-goal strategy treats the condition with the longest distance to goal. This can be considered "tackling the hardest problem" at every visit.

Closest-to-goal. The opposite of farthest-from-goal, this strategy selects the condition with the shortest distance to goal. In other words, this strategy pursues "low-hanging fruit". If incentives are set up to reward physicians for each condition for which a patient is at goal, this type of strategy would try to maximize the reward.

Risk-Based Strategies

Risk-minimizing to goal. Using a risk model, a strategy can incorporate the amount of reversible risk for a given condition by calculating a patient's current risk and then calculating the patient's risk if that condition were at goal. The difference between the two values is an estimate of the reversible risk for that condition and can be used as a

form of distance-to-goal metric. Reversible risk must be recomputed after each treatment because treating one condition changes how much risk reduction can be obtained by treating other conditions subsequently. The condition which, when treated to goal, results in the greatest risk reduction, will be selected by a farthest-from-goal strategy. Figure 9 displays the treatment path chosen by reversible risk distance-to-goal prioritization (for the same patient shown in Figure 8).

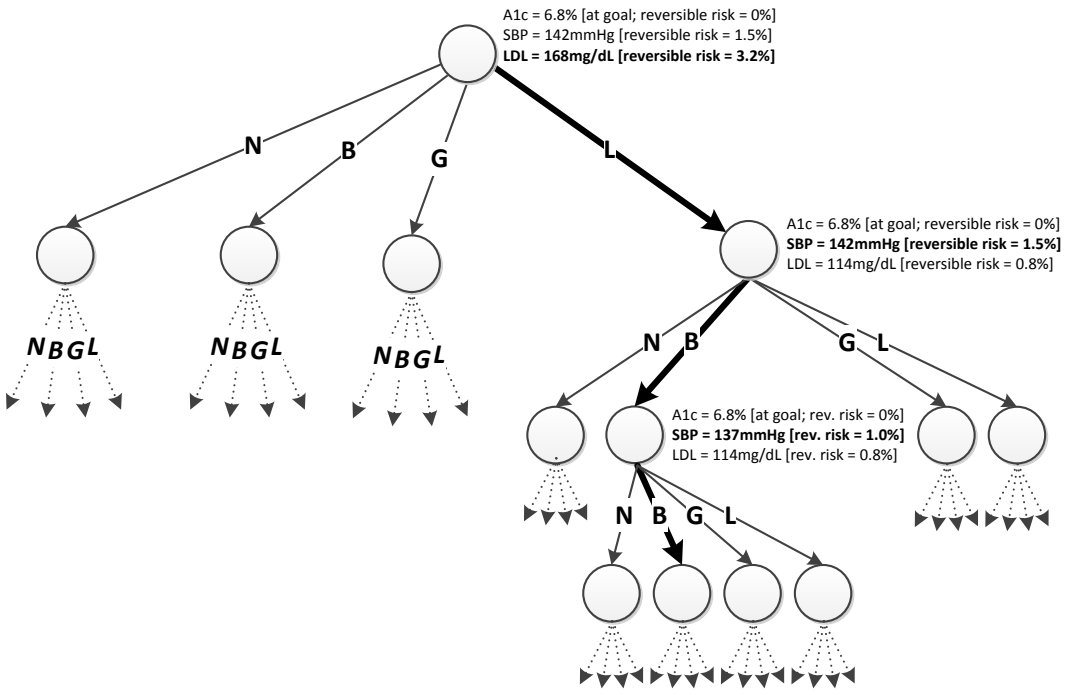

Figure 9. Example of a treatment path resulting from a risk-reducing strategy

Risk-minimizing at each move. This is a feedforward risk-minimizing strategy that estimates the expected risk reduction for different moves and selects the move that results in the greatest expected reduction. This strategy requires a strong mental model that

includes expectations for treatment moves. In addition, it requires access to a risk prediction model, such as the UKPDS or Framingham equations. For each condition for which a patient is above goal, the strategy computes (a) the patient's current risk and (b) the expected risk after making a treatment move for this condition. Subtracting (b) from (a) gives the expected reversible risk reduction for a move, and the strategy selects the move with the greatest expected reversible risk reduction. For example, imagine a patient with an estimated reversible risk reduction of 2% for making a treatment move on glucose and 1% for a blood pressure move. In this case, glucose would be selected for treatment.

In Table 7, the prioritization strategies are summarized in terms of what information the strategy requires, what type of computation is involved, and whether feedback or feedforward control is employed. The computations are described conceptually as follows. X represents the current value for a given patient state variable, for example, SBP. $G(X)$ represents the goal value for that variable, e.g., 130mmHg for SBP. $E[X]$ represents the expected change in the patient state variable X if a treatment move is made for this variable. For example, if X is SBP, then $E[X]$ represents the expected for a blood pressure medication move, e.g., -5mmHg. Consequently, $X - E[X]$ represents the expected SBP after one treatment move. $E[X]$ is used conceptually, its complexity depends on the forward model of a given strategy. For example, in the simplest case, the expectation is the same for every move, e.g., for any blood pressure medication prescription or does increase, the expected drop is 5mmHg. A more sophisticated forward model could contain a different expected drop for each drug class and depending on the

current dose. Any strategy that uses $E[X]$ must be feedforward strategy because its decision making depends on predictions about the patient state. *pct(X)* returns the percentile of the given value in the distribution of that patient state variable. For example, if X is SBP and the current patient has 160mmHg SBP, *pct*(X) returns .94 percentile, i.e., this patient is in the 94th percentile for systolic blood pressure.

argmin X: <expression> returns the condition for which <expression> has the smallest value; analogously, *argmax* returns the condition for which <expression> has the largest value. For example, *argmin* X: $X/G(X)$ returns the condition with the smallest percentage distance to goal $X/G(X)$.

Table 7. Conceptual description of prioritization strategies

Strategy	Required Information	Computation	Control
Serial treatment strategies			
Serial treatment	Patient state	X > G(X)	Feedback
Distance-to-goal strategies			
Closest-to-goal (%)	Patient state	argmin X: X/G(X)	Feedback
Farthest-from-goal (%)		argmax X: X/G(X)	Feedback
Closest-to-goal (pct)	Patient state; distributions	argmin X: pct(X)	Feedback
Farthest-from-goal (pct)	of A1c, SBP, LDL	argmax X: pct(X)	Feedback
Closest-to-goal (ex)	Patient state; expectations	argmin X: (X-G(X))/E[X]	Feedforward
Farthest-from-goal (ex)	for G, B, L moves	argmax X: (X-G(X))/E[X]	Feedforward
Risk-based strategies			
Risk-minimizing to goal	Patient state; risk estimate	argmax X: R(X)-R(G(X))	Feedback
Risk-minimizing at each move (ex)	Patient state; risk estimate, expectation for G, B, L moves	argmax X: R(X)-R(X-E[X])	Feedforward

Benchmark Strategies

No additional treatment. This reflects the patients not receiving any additional treatment for the timeframe of the study (two years). The Simcare patient model

calculates the resulting effects from disease progression. Comparing to this strategy demonstrates the benefit of treatment.

Optimal (post hoc). By creating a copy of a given patient at every visit and simulating each treatment choice, a tree as shown in Figure 10 can be constructed. Each node represents a patient state and the number in each node represents the risk associated that state. At the lowest level of the tree, the node with the lowest risk (highlighted in Figure 10) can be identified post hoc. This represents the lowest risk that could be obtained from prioritization within the constraints described above. The path leading to this node represents the optimal sequence (illustrated by the thick arrows) for this patient. This allows the comparison of prioritization strategies to the best outcome that could hypothetically obtained from prioritization.

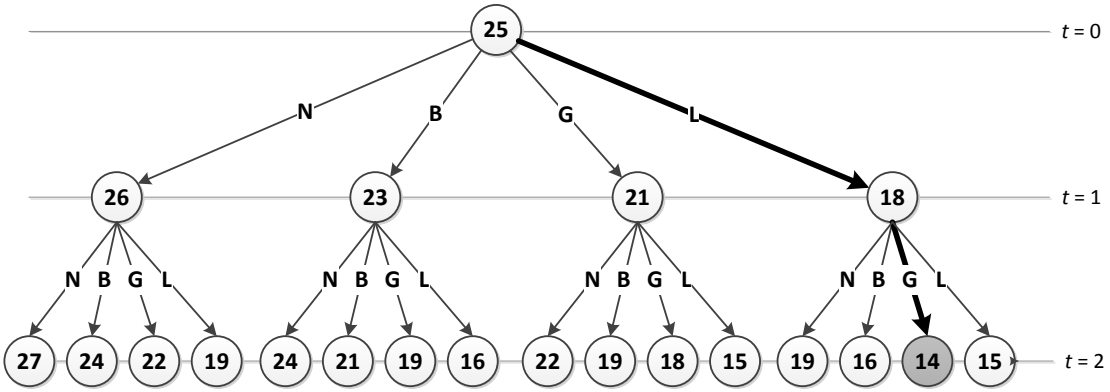

Figure 10. Tree showing risk in every node and highlighting optimal path

4.1. Methods

To examine the effects of different prioritization strategies, the Simcare diabetes patient model described in Section 3.2 is employed. We model the prioritization of three of the most common co-morbidities associated with diabetes patients: hyperglycemia, hypertension, and hyperlipidemia (Mazze et al. 2005). The health states resulting from treatment according to different prioritization strategies were determined using a clinical simulation model of T2DM patients. We focus on cardiovascular adverse events, namely myocardial infarction (MI) and stroke, as calculated using the UKPDS risk model described below (Stevens et al. 2001). The methodology scales to incorporating additional conditions and adverse event categories.

The simulation assumes patient visits every 90 days for four visits over one year of treatment. The simulated physician must select a treatment option at each visit unless all three conditions are at goal, therefore, each visit results in treatment intensification for patients not at goal.

4.1.1. Population

A population of synthetic patients is created based on a real population of 4,734 T2DM patients. These individuals were selected from the 18,356 T2DM patients described in 4.1.1 based on their need for prioritization. Prioritization can only be useful for patients with more than one condition above actionable thresholds for treatment. If a patient is above goal for a single condition, all prioritization strategies described above

would treat that condition, therefore, no difference would be found between the strategies for that patient.

Table 8. Population characteristics: synthetic patients at initialization (N=4,734)

Characteristic	Mean	SD
Irreversible risk factors		
Age	56.8	15.0
Female (%)	48.9	
Reversible risk factors		
BMI (kg/m^2)	33.8	8.6
Hemoglobin A1c (%)	9.1	1.9
Systolic blood pressure (mmHg)	138.3	18.3
Low-density lipoprotein LDL (mg/dl)	141.2	33.1
High-density lipoprotein HDL (mg/dl)	42.4	12.5
Triglycerides (mg/dl)	191.0	116.6
Current smoker (%)	5	
Co-morbidity profile		
A1c and SBP > 80th percentile (%)	16.0	
A1c and LDL > 80th percentile (%)	47.5	
SBP and LDL > 80th percentile (%)	22.4	
A1c, SBP, and LDL > 80th percentile	14.1	
Treatment at initial visit		
On hyperglycemia medication (%)	65.3	
On hypertension medication (%)	51.6	
On lipid medication (%)	32.7	

Patients that are currently treated for a condition are initialized with the medications previously prescribed. Prioritization strategies take into account treatment moves that have already been made.

4.1.2. Risk Estimation

Cardiovascular risk for given a patient health state is estimated using the United Kingdom Prospective Diabetes Study (UKPDS) Risk Engine, which provides a set of equations to estimate a patient's 10-year risk of experiencing fatal or non-fatal CHD or stroke events (Stevens et al. 2001). The coefficients were estimated based on data from

4540 patients who participated in the United Kingdom Prospective Diabetes Study (Turner et al. 1991; Stratton et al. 2000). For the purpose of this study, the risk of CHD events and strokes are combined into a single risk number for a *general cardiovascular event* which is used to guide prioritization.

Each prioritization strategy has been implemented as an algorithm that takes patient state as input and outputs which condition to treat. The selected condition is treated using the regimen discussed in Section 3.4. Each strategy is applied to the 4,734 synthetic patients for four visits in a one year timeframe. After this treatment, each patient's CHD and stroke risk are calculated using the UKPDS Risk Engine and combined to obtain the general cardiovascular event risk for that patient for a ten-year timeframe. After this has been completed for all patients, Monte Carlo simulation is used to generate events based on these risk estimates and determine the population event rate (for details, refer to Appendix A).

Two strategies can be compared by considering their difference in event rate. Standard errors for these differences are estimated using 40 bootstrap samples from the simulated population and two-sample t-tests are performed at a two-sided significance level of 0.05. The results are presented below.

4.2. Results

The base case uses the population of 4,374 T2DM synthetic patients in need of prioritization of care. We assume 4 visits with 90 day intervals. Prioritization strategies which rely on expectations, such as risk-based prioritization, use the following

expectations derived from literature and empirical observations in the patient population: A single glucose medication adjustment is expected to lower A1c by .375%, a single blood pressure move is expected to lower SBP by 5mmHg, and a single lipids move will reduce LDL by 30% if it is the first time lipids are treated or 10% any other time.

Table 9 displays the results of all strategies for comparison. *10 year event rate after treatment* is the estimated general cardiovascular event rate after one year of simulated treatment according to the prioritization strategy for a given row. The proportions of *glucose moves*, *blood pressure moves*, and *lipids moves* represent how prioritization strategies divided up their treatment efforts, i.e., if a strategy only made glucose moves, this number would be 1. The *mean additional cost of treatment* is the additional treatment cost for one year above the cost of the treatment that the patient has already received by the time of the initial visit. It reflects the treatment costs that are a direct result of the choices of the prioritization strategy. Even "no additional treatment" would have a positive total cost of treatment since we did not simulate discontinuing existing treatment, however, it does not incur any additional treatment costs because it does not make additional treatment moves.

Table 9. Results for all strategies

Strategy	10 year event rate after treatment	Proportion of glucose moves	Proportion of BP moves	Proportion of lipids moves	Mean additional cost of treatment ($)
Benchmarks					
(-) No additional treatment	.320 −	.000	.000	.000	0
(#) Random Choice	.294	.391	.284	.325	521 *
(+) Optimal (post hoc)	.258 +*	.406	.101	.493	506 *
Serial treatment strategies					
(A1) Serial BGL	.320 −	.432	.545	.023	543
(A2) Serial BLG	.298	.160	.545	.295	422
(A3) Serial GBL	.319 −	.768	.209	.023	616
(A4) Serial GLB	.313	.768	.074	.158	592
(A5) Serial LBG	.264 +*	.155	.352	.493	359
(A6) Serial LGB	.258 +*	.435	.072	.493	388
Distance-to-goal strategies (percentage metric)					
(B1) Closest-to-goal (%)	.316 −	.397	.471	.132	521 *
(B2) Farthest-from-goal (%)	.286	.509	.194	.296	513 *
Distance-to-goal strategies (percentile metric)					
(C1) Closest-to-goal (pc)	.312	.424	.333	.243	359
(C2) Farthest-from-goal (pc)	.302	.768	.074	.158	592
Distance-to-goal strategies (expected number of moves metric)					
(D1) Closest-to-goal (ex)	.296	.285	.412	.303	506 *
(D2) Farthest-from-goal (ex)	.313	.612	.283	.105	589
Risk-based strategies					
(R1) Risk-minimizing at each move	.259 +*	.397	.110	.493	389
(R2) Risk-minimizing to goal	.276 *	.537	.206	.257	535

+ significantly lower than random choice, − significantly higher than random choice

* not significantly different from optimal

Feedback vs. Feedforward. Three strategies that use feedforward control are represented in the results shown in Table 9: (D1), (D2), and (R1). (R1), which calculates the risk benefit for a move and selects the move with the greatest expected risk benefit, achieves performance that is not significantly different from optimal performance (+). The feedforward distance-to-goal strategies (D1) and (D2) perform not significantly different from random choice, indicating that feedforward control alone is not sufficient to achieve good performance. However, feedforward control paired with risk information

(in strategy (R1)) results in making the optimal lipids moves and ends up with a very similar profile of moves as optimal treatment. The event rate obtained by (R1) is almost identical to the event rate from optimal treatment (0.1% difference); (R1) and (+) are not significantly different with $p < .01$. In comparison, (R2) uses risk information with feedback control and results in an event rate that is 1.8% higher than (+) and at the boundary of the 95% confidence interval for (+).

Distance-to-goal Metrics. There is no significant difference between random choice (#) and the strategies that use percentile (C1) and (C2) or expected number of moves (D1) and (D2) as a distance-to-goal metric. The strategy (B1), which treats closest-to-goal according to a percentage metric, performs significantly worse than random choice, indicating that it consistently selects moves that have minimal impact on risk. Random choice performs better because it selects moves that impact risk by chance. (B2), which uses the same percentage metric as (B1) but treats the condition farthest-from-goal, performs significantly better than (B1).

Risk information. Both strategies that explicitly incorporate risk information (R1) and (R2) perform significantly better than random choice and are not significantly different from optimal treatment, making a strong case for the use of risk information. The distance-to-goal strategies that do not use risk information all perform not significantly different from or worse than random choice.

Serial treatment. The two serial treatment strategies that bring lipids to goal first (A5) and (A6) are among the top performers, which are not significantly different from

optimal treatment (+). (A5), (A6) and (+) all make lipids moves first, after which (A6) closely tracks (+) by focusing on glucose while (A5), by design, treats blood pressure next. The two serial treatment strategies that bring lipids to goal last (A1) and (A3) are among the worst performers, which are significantly worse than random choice (#) and not significantly different from "no additional treatment" (-). If a strategy commits to treating a condition to goal, it is crucial that it is a strategy for which each move has an impact on risk. Only 2.3% of all the moves made by (A1) and (A3) are lipids moves, and lipids moves are very effective in reducing risk. Random choice (#) outperformed (A1) and (A3) because it treated lipids at 32.5% of its encounters.

Top performers. Four strategies emerge as top performers that are not significantly different from optimal treatment: the two serial treatment strategies that bring lipids to goal first (A5) and (A6), and the two risk-based strategies (R1) and (R2). The success of the serial treatment strategies can be explained in terms of the environment. In this version of the prioritization of care problem, a natural ordering seems to exist in the power of different classes of prioritization moves to impact risk. For the investigated population, a single lipids move generally lowers cardiovascular risk more than a single glucose move, which in turns generally lowers risk more than a single blood pressure move. However, the difference between blood pressure and glucose is not as pronounced as the difference between glucose and lipids. As a result, a feedback strategy that checks whether a condition is above evidence-based goals and treats conditions to goal in this same order can achieve close to optimal performance. In fact, (A6) and (R1) choose the same treatment sequences in the majority of cases. In other words, the treatments

resulting from the simple decision rule of treating serially (A6) appear as though they were chosen by a strategy that was aware of the risk implications of each move, such as (R1).

Cost of treatment. In terms of treatment costs, the best performing strategies (i.e., those with the lowest cost) are, in order, (A5), (C1), (A6), (R1), and (A2). The worst-performing are, starting at the strategy with the greatest cost of treatment, (A3), (A4), (C2), and (D2). The serial treatment strategies have the greatest variance in cost. Serial LBG (A5) has lowest cost of any strategy in the experiment, and serial GBL (A3) the highest. Lipids moves are generally inexpensive (using generic drugs) compared to glucose and blood pressure moves, especially for patients who are already on more than one medication for glucose of blood pressure treatment at the beginning of the study.

Cost effectiveness. In terms of efficacy, four strategies dominate: (A5), (A6), (R1), and (R2). Three of these also dominate in terms of treatment cost: (A5), (A6) and (R1). Their cost-effectiveness ratios (cost per prevented event) were tested and are not significantly different from each other. Therefore, these three strategies are the most cost-effective. This is, in part, due to the fact that lipids moves are not only very effective at reducing risk, they are also relatively inexpensive compared to blood glucose and blood pressure moves, resulting in (A6) and (R1) having considerably lower costs than (R2) which achieves similar performance in terms of risk reduction, but does so by focusing more on blood glucose and blood pressure treatments instead of lipids.

4.2.1. Sensitivity Analysis: Expectations

The feedforward prioritization strategies discussed above rely on expectations of the effect that a treatment move will have, represented as $E[X]$ in Table 7. We study the effect of varying these expectations to determine the sensitivity of our results. Perfect expectations are derived by giving the decision strategy full advance knowledge of the effect an action will have in the Simcare patient model, for example, taking into account varying dose response and time effects (McCabe 2012).

Table 10. Expectations for feedforward strategies (forward model)

Variable	Simple expectations	Advanced expectations	Perfect expectations
	Single point estimates	Derived from literature and empirical observations in the patient population	To provide a benchmark, the exact effect of treatment is calculated by the simulation model and fed into the prioritization strategy. The result is a prioritization strategy that can perfectly anticipate the effects of its action by having complete knowledge of the Simcare patient model (McCabe 2012).
A1c	-0.5 %	- 0.375 %	
SBP	-5 mmHg	-5 mmHg	
LDL	20% drop	30% drop first move, 10% drop following moves	

Table 11. Results for risk-minimizing at each move prioritization (R1) with different expectations

Strategy	10 year event rate after 4 moves	10 year event rate after 8 moves
Simple expectations	.306	.240
Advanced expectations	.259 +*	.220 *
Perfect expectations	.258 +*	.219 *

+ significantly lower than random choice, − significantly higher than random choice
* not significantly different from optimal

The performance of risk-based strategies is sensitive to the expectations that are assigned to given moves, underscoring the importance of estimating an appropriate model

for expectations prior to implementing a feedforward strategy. A successful "advanced expectations" model with the four rules stated in Table 4 performs statistically as well as a perfect expectations model that contains specific rules for every medication and dose combination. The predictions made by the "advanced expectations" model are sufficiently close to the predictions made by a perfect expectations model that they result in the same moves for virtually all cases. The perfect expectations model produces considerably more precise predictions; however, this level of precision is not required for the problem environment given here.

4.3. Discussion and Conclusion

This study has demonstrated that risk-based prioritization as well as serial treatments focusing on lipids first (LGB and LBG) are effective and robust strategies to prevent complications. In one year of treatment, these strategies can lower event rates as much as an idealized optimal strategy that has perfect advance knowledge of what its actions will accomplish. The strategies are particularly beneficial to guide choices for early visits with complex patients; after more visits, the need for prioritization diminishes and the performance of different strategies begins to converge. As discussed above, the good performance of serial LGB and LBG can be explained as a function of the relative power (to reduce risk) of the moves associated with lipids, glucose, and blood pressure drugs. Both strategies can be applied instead of risk-based strategies if visits are infrequent (every 90 or more days) and as long as there are no changes in the available moves. For example, if a new drug was introduced that could lower blood pressure by 30 points with a single move, the performance of the serial strategies might no longer be near-optimal. A

feedforward risk-based strategy, on the other hand, could take such a new expectation into account and still retain its performance.

In the simulation experiment, this convergence can be observed after two years of treatment. This is a function of intensifying treatment every 90 days if necessary, a condition that often does not hold in the real world. If there are fewer visits or longer intervals between visits resulting in intensification, the benefits of an appropriate prioritization strategy are even greater because the convergence is delayed. If more visits (treatment opportunities) were permitted, strategies that are able to take advantage of shorter treatment windows (i.e., feedforward strategies) would be expected to achieve risk reduction sooner than feedback-based serial treatment or distance-to-goal strategies.

Closest-to-goal strategies perform significantly worse than random choice. These strategies fail because their mental model is inaccurate – they are focused on getting to goal by addressing "low hanging fruit" first rather than accurately assessing the risk impact of treating conditions that are farther from goal. The resulting performance is, in part, due to the very limited risk benefit from treating conditions very close to goal. For example, a simulated 55-year old Caucasian female non-smoker with a history of atrial fibrillation, SBP of 160mmHg, and LDL of 120mg/dl has a stroke risk of 25.6% according to the UKPDS Risk Engine. For this patient, treating lipids (the "low-hanging fruit") reduces her stroke risk to 24.4%. In contrast, leaving LDL at 120mg/dl and treating blood pressure instead (which is farther from goal) reduces the patient's stroke risk to 18.9%. To achieve the goal of reducing risk, bringing conditions that are already

close to goal below goal may not be as beneficial as initiating or intensifying treatment on a different condition. This may have important implications for the design of physician incentives. As discussed above, if incentives reward maximizing the number of patients at goals, a closest-to-goal strategy would maximize reward collection by making treating decisions that focus on "low-hanging fruit".

There are several limitations to this study. Some prioritization strategies, especially those with strong feedforward models, are severely limited by the constraints presented for this problem. They can gain considerable power if these constraints are relaxed, for example, by allowing more than four visits or allowing shorter visit intervals. Strong prioritization strategies can then schedule additional visits or use more than one medication at a given visit in order to reduce risk faster. In Study 2, these constraints will be relaxed to allow for strategies to use their full power, which also introduces the possibility of overtreatment.

This study is focused on prioritization strategies to prevent cardiovascular events. More comprehensive prioritization strategies might also consider other complications, such as blindness, amputations, and kidney failure. In addition, they might want to take into account other reversible risk factors, such as weight, smoking status, the use of aspirin in high risk patients, and additional treatment options such as poly-pills and lifestyle changes. The study presented here makes an initial contribution to demonstrate the efficacy of prioritization strategies. The methodology can readily be expanded to take additional variables into account. To do so, a risk model that considers the additional

reversible risk factors is required, and accurate expectations for the treatment effects of these factors must be developed.

Moreover, it is not clear whether two patients with a blood glucose of 8% A1c have equal risk if one patient is untreated and the other patient is on three classes of blood glucose medications and had an untreated A1c of 11% (Chalmers & Cooper 2008). The UKPDS risk engine only considers the current patient state and would treat these two individuals identical. However, if the path by which a patient arrived at the current state matters, this should be taken into account for risk-based prioritization. It has been shown (and is accounted for in the UKPDS risk model) that such a legacy effect exists for smoking, a former smoker has higher risk than an individual that never smoked (Stevens et al. 2001). If legacy effects exist for other treatments, a risk model that takes patient history into account will need to be developed.

Furthermore, in order to achieve the goal of preventing adverse events and complications, it is critical that the risk model used for prioritization is an accurate estimator of the probability of events. Otherwise, while the estimated risk number may be reduced by treatment, the number of prevented events may be lower than expected.

In conclusion, this study demonstrates the efficacy of prioritization strategies to prevent future complications even when the number of treatment opportunities (visits) is very limited. The findings presented here may be informative for patients to understand the risk implications of treatment choices and to guide treatment choices for complex patients, especially during their first few visits.

5. Study 2: Determination of means to identify and compensate for systematic errors in dynamic decision strategies

In Study 1, specific prioritization strategies were examined in a constrained version of the prioritization of care problem. In Study 2, these constraints are relaxed to reflect the variety of possible treatment choices in the real world more accurately. Given these relaxed constraints, the performance of existing strategies is investigated and an approach is introduced to improve existing strategies with respect to their efficacy, costs, and cost-effectiveness.

5.1. Background

Brehmer (1992) argues that some of the aspects of dynamic problems, such as delay of feedback and lack of transparency, reduce the performance of feedback control but can be addressed by feedforward strategies that make predictions about the future state of the system. For example, if delay of feedback is anticipated, a decision maker can rely on the prediction of what his action will accomplish instead of either overreacting because no feedback is currently present or not reacting at all, i.e., waiting for the delay to pass. However, if it is not easy to inspect a system because one can only observe the system's changes in response to actions, how can a mental model suitable for feedforward control be formed?

To identify opportunities for improvements in decision strategies, we develop an approach based on machine learning and control theory named **Procedo** - **Pr**ediction **o**f

control **e**rrors in **d**ynamic c**o**ntexts. Procedo is a systematic approach to (a) determine conditions under which a given strategy fails and (b) improve this strategy under these conditions. Specifically, we state the determination of conditions which are predictive of failures as a data mining question: what information is contained in the state of the system at a given point in time that is indicative of a control failure occurring at a later time? This enables the application of data mining techniques, such as decision tree induction, to predict future failures based on present conditions. A decision strategy can then be improved for the conditions under which failures are predicted. Using domain-specific knowledge of outcome and cost trade-offs, Procedo evaluates different strategy modifications for the purpose of selecting the most cost-effective one, which then is incorporated into an improved decision strategy.

The Procedo approach presented in this study proposes that feedforward control can be introduced (to existing strategies that use feedback control) or augmented (for strategies already using feedforward control) by recognizing patterns of success or failure for a given decision strategy in empirical data, using these patterns to predict future failures and making appropriate adjustments in anticipation of failures. Prior literature combining dynamic decision making with the process control framework pointed out that control theory stems from continuous process control and cannot be readily applied to the discrete nature and pattern recognition aspects of decision making in dynamic environments (Bainbridge 1981; Brehmer 1992). For this study, we propose to pair the control theory framework with a data mining technique in order to incorporate pattern recognition mechanism into process control, specifically into the mental model, with the

aim of improving performance. Data mining is used to identify conditions under which a decision fails to bring a system to a desired goal state in a specified amount of time, brings a system to a goal state faster than it safely can be brought to goal, or consumes excessive resources (as measured by cost). These instances are referred to as *control failures*. This study is focused on identifying control failures that are a result of *errors*. A failure may occur either because a strategy failed to take action or took insufficient action, referred to as *error of omission*, or because it took excessive or harmful action, referred to as *error of commission*.

Modern processes are frequently monitored using information systems which record large amounts of information about the process. Such data can then be used to identify failure conditions in a process and make adjustments for future runs. Recording process information frequently results in very large data sets that are not necessarily tractable to analysis by individuals or usefully interpreted using common statistical techniques such as linear regression or t-tests. For example, even very small differences can become statistically significant when making comparisons on data sets with millions of records. Data mining methods have emerged to provide ways to undercover interesting patterns or learn new information from very large data sets (Friedman 2001).

Sadoyan et al. (2004) have shown that data mining offers potential for improvement of process control. They have successfully developed a new data mining algorithm based on rough set theory specifically tailored towards improving a manufacturing process. The objective of this study is to develop a general, domain-independent approach that

employs commonly used and proven data mining techniques to dynamic decision strategies.

Traditional process mining approaches attempt to discover and document the nature of processes that generate an observed chain of events (van der Aalst & Weijters 2004). In contrast, the approach proposed in this study aims to control a system by means of a decision strategy and uses observations and available actions to modify this strategy in order to increase its performance without requiring the discovery of a full model of the system being controlled. This makes Procedo valuable in processes that are difficult to analyze but that react consistently to interventions. For example, disease processes are difficult to analyze, but can be affected through certain medications. However, some patients may have factors, such as genetics, that make some medications ineffective for them. Given enough relevant observations, Procedo can identify such nuances, allowing the formulation of customized, more effective decision strategies.

Procedo employs established data mining techniques to recognize patterns which result in errors, i.e., under-performance or excessive, wasteful, and, therefore, unnecessarily costly actions. Therefore, it must be able to provide a classification between successful outcomes and failures. While many classification techniques are available, such as neural networks, support vector machines, naive Bayesian classifiers, and decision tree learning (Tan et al. 2006), the goal of Procedo is to output a set of conditions for which a decision strategy can be improved. We have chosen decision tree learning algorithms, in particular the C4.5 algorithm (Quinlan 1987), specifically for its

ability to partition instances to outcome classes based on the most informative distinguishing criteria. C4.5 provides a set of *readily interpretable* conditions which can be incorporated into a decision strategy using Boolean logic. Interpretability is a desirable characteristic for many real world dynamic decision contexts; for example, physicians would likely be reluctant to take treatment action based on a "black box" prediction technique (Avins 2010), such as a neural network. Furthermore, the interpretable nature of a decision tree node provides feedback to the decision maker that can guide the adjustment of the decision strategy. The characteristics of the instances in which the strategy performs poorly are human-readable and easy to encode as IF-THEN statements in computer programs or database selection criteria.

5.2. Prediction of Control Errors in Dynamic Contexts: An Overview of the Proposed Approach

Consider a system that is to be controlled in a dynamic context. For example, this system may represent a chronic disease patient. Consistent with Seborg et al. (1986), the state of the system is fully represented by the tuple (c, m, d). c is the set of controlled state variables; the goal of process control is to bring these to desired values. For example, for a diabetic patient, blood sugar is a controlled variable. m represents the set of manipulated state variables that can be affected by the decision maker and which in turn affect c. The amount of medications that affect blood sugar, e.g., insulin, is an example of a manipulated variable. d denotes the set of disturbance state variables that affect c but cannot be affected by the decision maker. An example of a disturbance variable is patient age, which correlates with increased blood sugar and affects the risks

of diabetes-related complications but cannot be affected by a physician. Figure 11 refines the process control model shown in Figure 1 to illustrate the interactions between the decision strategy and the variables of the system being controlled.

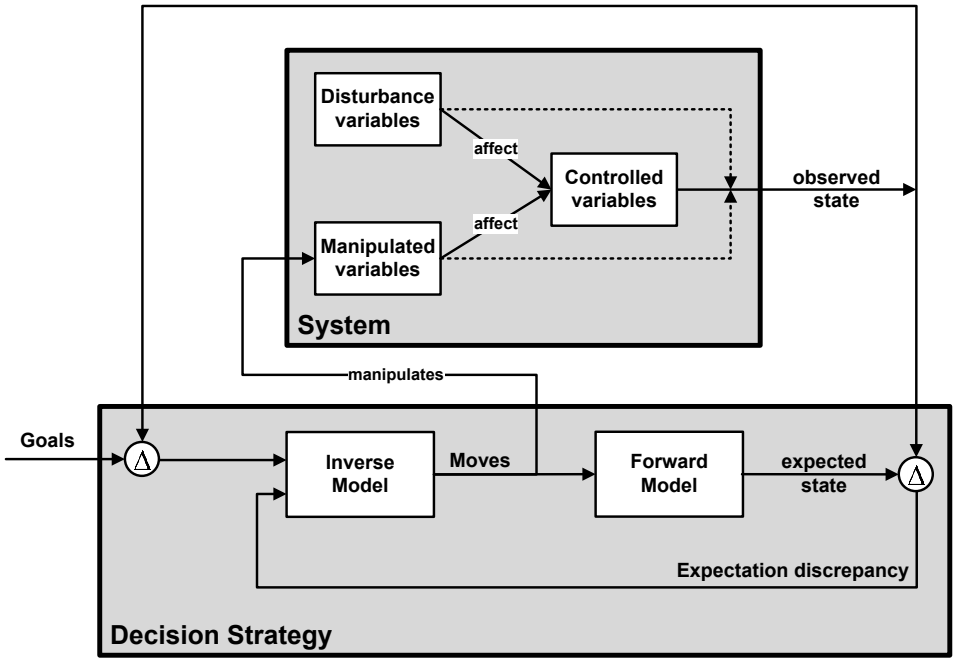

Figure 11. Process control model (from Figure 1) marking boundaries between decision strategy and system, and decomposing the state within the system into controlled, manipulated, and disturbance variables

In dynamic decision contexts, the system changes autonomously over time (Edwards 1962). The state of the system at a specific point in time t is denoted by (c_t, m_t, d_t). The effect of time on the system can be an unknown function that modifies the state of the system for the passage of the amount of time Δ. This function is represented as $T(c_t, m_t, d_t, \Delta)$. If the system is uncontrolled, its state is determined by the following equation:

$$(c_{t+\Delta}, m_{t+\Delta}, d_{t+\Delta}) = T(c_t, m_t, d_t, \Delta).$$

The purpose of controlling the system is to bring the controlled state variables c to designated goals. The decision maker's task is to manipulate m at discrete points in time so that, with the passage of time, the system will reach a goal state. For this purpose, the decision maker adopts a decision strategy, a way of making and executing decisions at every decision point. In this study, we consider the cases for which the decision maker incorporates feedback, i.e., decision strategies that take the state variables (c, m, d) of the system to be controlled as inputs and output new values for the manipulated variables in m in order to bring the system towards a goal state. A strategy s is represented as a mapping from a given set of state variables (controlled, manipulated, and disturbance) to a new set of manipulated variables:

$$s: C \times M \times D \rightarrow M.$$

An application of the strategy $m' = s(c, m, d)$ returns m' representing the values of the manipulated variables after they were changed by the strategy. Feedback is necessary in order to assess whether the system has reached a goal state. The mechanism to run a strategy is described in Algorithm 1.

Algorithm 1 run-strategy

Input $\mathbb{I}$: a set of process instances providing information baed on which a strategy is improved
Input T: amount of runtime after every iteration of improvement
Input n: version number of the decision strategy used by the process
Output $\mathbb{H}$: history, i.e., the states of each instance at every point in time during the process

$\mathbb{H} \leftarrow \{(c_{i,0}, m_{i,0}, d_{i,0})\}$
for $i \in \mathbb{I}$ **do**
    $t \leftarrow 0$
    while $t \leq T$ **do**
        $m'_{i,t} \leftarrow s_n(c_{i,t}, m_{i,t}, d_{i,t})$
        $(c_{i,t+\Delta}, m_{i,t+\Delta}, d_{i,t+\Delta}) = T(c_{i,t}, m'_{i,t}, d_{i,t}, \Delta)$ // After taking action, wait for amount of time Δ
        $\mathbb{H} \leftarrow \mathbb{H} \cup (c_{i,t+\Delta}, m_{i,t+\Delta}, d_{i,t+\Delta})$
        $t \leftarrow t + \Delta$
    end while
end for
return $\mathbb{H}$

Algorithm 1 requires three inputs: a set of instances $\mathbb{I}$ to which the process is applied (e.g., widgets in manufacturing, patients in healthcare), the amount of time T for which the process is run, e.g., one hour or one year, and the version of the strategy that is run (in the initial scenario n=0). The meaning of n will become apparent shortly. Algorithm 1 returns the observed history $\mathbb{H}$ of the process, i.e., the complete state (of controlled, manipulated, and disturbance variables) at every point in time t for each instance $i \in \mathbb{I}$.

The initial decision strategy (version n=0) refers to the original strategy of making decisions based on the state (c, m, d) about how to modify the set of manipulator variables m in order to control the system. The goal of Procedo is to define successive versions of the decision strategy, each version representing an improvement over the

prior one. Due to the dynamic nature of the problem domain, strategies typically are not analytically tractable, that is, they cannot be improved using traditional optimization techniques, linear programming, or numerical gradient descent approaches (Nocedal & Wright 2006). Therefore, Procedo serves as a heuristic approach to improve decision strategies based on empirical observations of outcomes and costs associated with a strategy, i.e., without knowledge of its inner workings.

In summary, Procedo is designed as an iterative process that attempts to improve an initial decision strategy, s_0, in terms of cost-effectiveness as measured by Φ (see Section 2.3.3). Improvements in cost-effectiveness can be accomplished (a) by improving outcomes, e.g., reducing the number of events, without increasing costs or (b) saving costs without lowering performance on outcomes. In every iteration of Procedo, improvements are added to given version n of a strategy to create candidates for the next version of the strategy s_{n+1}. Each candidate strategy is evaluated using Φ, and the candidate with the best Φ is chosen as s_{n+1}. Improvements are added so that improved strategies resulting from the application of Procedo are always of the form depicted in Figure 12 and specified in Algorithm 2. Every new version $n > 0$ of the strategy forms a *wrapper* around the previous strategy $n - 1$ and will only modify an action if a failure is predicted. For example, strategy version $n = 1$ will take additional action *only if* the initial strategy (version $n = 0$) is predicted to fail; otherwise the initial strategy is executed. Thus, Procedo applies only minimal additional effort, i.e., only when it is necessary to prevent failures (as will be illustrated further below), which is an important aspect of the proposed approach. It is also important to point out that every version of the

strategy takes the same inputs and produces outputs of the same format, making this approach is independent from the initial strategy. The original strategy can be a "black box" and yet Procedo can design incremental improvements to wrap around it, transparent to the user of the decision strategy since the interface (inputs or outputs) does not change.

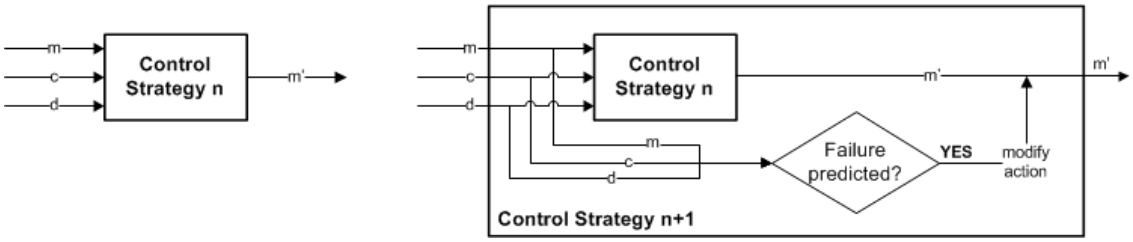

Figure 12. Decision strategy versions *n* (left) and *n+1* (right), illustrating the "wrapper" approach to strategy modification

Algorithm 2 strategy s_{n+1}

Input (c, m, d): The tuple of controlled, manipulated, and disturbance variables describing the current state of the system being controlled	
1: $m' \leftarrow s_n(c, m, d)$	// Recursively call previous iteration control strategies
2: **if** predict-failure$_n(c, m, d)$ **then**	
3: $m' \leftarrow$ modify-action$_n(m')$	// Control failure predicted: modify action
4: **end if**	
5: **return** m'	

This completes the high-level overview of the different steps of Procedo as illustrated in Figure 13. In the following section, each step is described in greater depth.

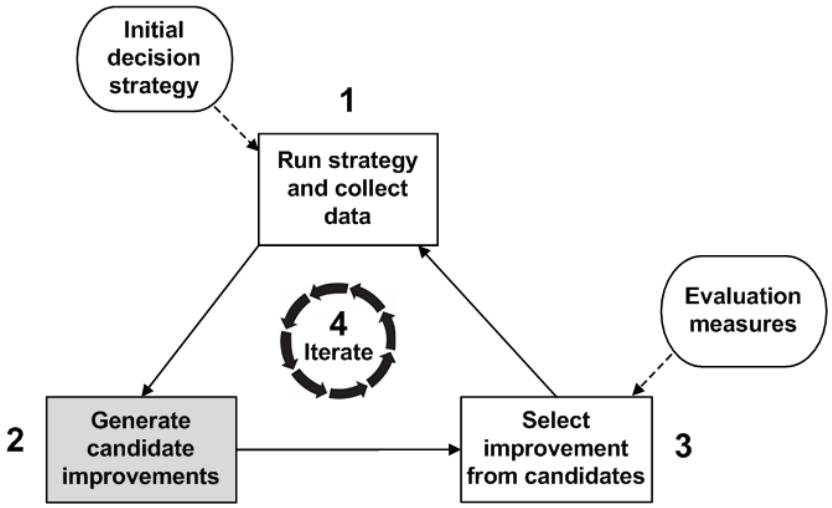

Figure 13. Overview of the Procedo approach

5.2.1. Generating Improvement Candidates

A key contribution of this study is the way in which improvements are generated. This step is shaded in Figure 13 and illustrated in more detail in Figure 14.

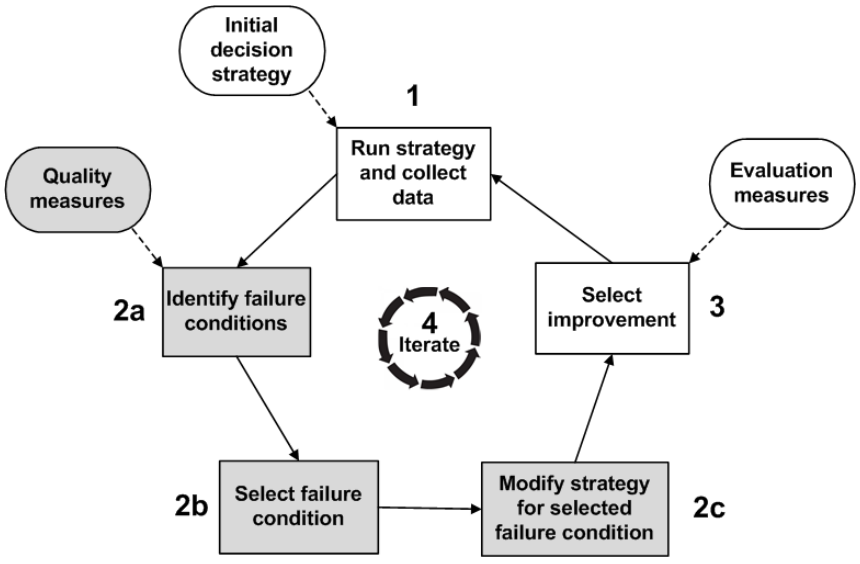

Figure 14. Detailed steps of the Procedo approach

To generate candidates, Procedo requires domain-specific quality measures to determine whether a control failure occurred, and if so, what the nature of the failure is. A failure can either be the result of insufficient action, which adversely impacts outcomes, or a result of excessive action, which adversely impacts cost. We refer to the former as failures of type *E*- and the latter as failures of type *E*+. Using decision tree learning, the conditions predicting these failures are identified. For the top k identified failure conditions, as ordered by accuracy and support, improvements are added to the decision strategy. Choosing a k smaller than the number of identified failure conditions limits the number of conditions for which improvements have to be generated and evaluated in order to increase the scalability of the proposed approach. For the purpose of illustrating the proposed approach in this study, we set $k = 1$.

5.2.2. Initial Decision Strategies

Before describing the specific steps in detail in Section 5.3, as a starting point for Procedo, an initial decision strategy must be selected. In addition, the criteria for assessing the quality of outcomes for the dynamic decision making process have to be defined.

The initial strategy s_0 modifies the set of manipulator variables m based on the current state of the system in order to control the system. Three initial strategies, *MIN*, *MAX*, and *EXP*, are provided for the diabetes care example. These strategies provide the order and dosages in which medications are described as well as rules governing the prescription of insulin and the frequency of office visits. Examples of actions available to

these treatment strategies are “prescribe 500mg Metformin” and “schedule a visit in 90 days”. The patient's blood glucose level is a controlled variable; the amount of medication prescribed to a patient is a manipulated variable.

MIN is a very weak strategy, informed by “usual care” literature, in need of augmentation (Phillips et al. 2001; Selby et al. 2003; B. R. Shah et al. 2005). *MIN* can be characterized as a weak feedback strategy that only makes moves after 180 days since the last move for a given condition, or after observing no further improvement in a given condition between two visits. MIN will only treat one condition at a given encounter and prioritizes by percentile farthest-from-goal. It represents a weaker version of the strategy “farthest-from-goal (pct)” presented in Study 1.

MAX is a very aggressive and costly strategy, informed by diabetes specialist guidelines (Mazze et al. 2005). *MAX* is a very strong feedback strategy that operates under the assumption that a move is 95% realized after 60 days and, therefore, may make another move on a condition after 60 days. As a default, *MAX* schedules visits every 30 days to bring patients to goal. Given its feedback nature and aggressive treatment, *MAX* has potential to commit errors of commission, e.g., excessive visits and overtreatment. *MAX* can treat two conditions at a given visit, making it more powerful than the strategies represented in Study 1, which were limited to choosing a single condition at a given encounter.

EXP is an effective strategy between the extremes presented by *MIN* and *MAX*, informed by observations of experts in clinical practice (Ramsey et al. 2010). *EXP*

represents a weak feedforward strategy that, as a default, schedules visits every 60 days. *EXP* attempts "soft landings", i.e., slowing treatment when a patient gets close to goal to avoid overtreatment. *EXP* treats a single condition at a given encounter, using expectation-based prioritization, comparable to the strategy "farthest-from-goal (ex)" in Study 1.

5.2.3. Quality Metrics

To determine whether a control failure occurred, a domain-specific quality measure q is required that specifies the conditions for insufficient performance, i.e., failures of type *E*-, and the conditions for failures of excessive or wasteful performance, i.e., failures of type *E*+. In chronic disease care, the goal of the treatment strategy is to bring a patient's blood sugar, blood pressure, and cholesterol (controlled variables) to evidence-based goals (defined in Table 2) in order to minimize the risk of complications or adverse clinical events, such as heart attacks and strokes. At the same time, excessive treatment may be harmful to the patient or waste resources of the healthcare system. Therefore, let τ_i represent the time when a patient reached evidence-based goals the first time during the course of treatment, and let T represent the required time by which the patient should be at goal. We can now define the following quality measure q for a strategy s and a given patient i.

$$q(s,i) = \begin{cases} E^+ & \text{if } 0 < \tau_i < T/2 \quad \text{(excessive performance)} \\ \emptyset & \text{if } T/2 \leq \tau_i \leq T \quad \text{(desired performance)} \\ E^- & \text{if } \tau_i > T \quad \text{(insufficient performance)} \end{cases}$$

This quality measure states that a treatment strategy is successful if a patient is at evidence-based goals from the beginning of treatment or arrives at evidence-based goals before the required timeframe, but not sooner than in half the required time. In the former case, the actions taken by the treatment strategy were insufficient (failure *E*-), in the latter, the actions were excessive (failure *E*+). *E*- failures are indicative of errors of omission; *E*+ failures may be the result of errors of commission. Applying modified actions instead may bring patients to goal or result in appropriate progress, respectively. In the chronic disease context, insufficient progress is often the result of errors of omission; for example, a physician may not have attended to a condition that needed to be treated because of other acute problems (Parchman et al. 2007). Additional treatment moves are needed to control patients. Excessive actions occur either in the form of over-treatment, which can adversely affect patients, or in the form of waste, for example, when unnecessary visits are scheduled.

5.3. Step-by-Step Description of the Proposed Approach using the Diabetes Care Example

The definition of the initial strategy s_0 and the function to identify failures **is-failure** with respect to which we wish to improve the strategy are prerequisites for Procedo. Our proposed approach, as represented in Figure 13, can then be run and will be explained step by step below.

5.3.1. Step 1: Run Process and Collect Data

Data is collected by applying **run-process** (discussed in Section 5.2) with the initial strategy s_0 to the system to be controlled or a model thereof. At given points in time, e.g., patient visits, the observed state is recorded to build history set $\mathbb{H}$. In the example, treatment of synthetic patients is simulated using the Simcare patient model, here denoted at PM. This model can be represented as follows: $X_{t+\Delta} = PM(X_t, A_t, \Delta)$, i.e., it takes patient state X_t at time t and the prescribed treatment actions A_t (e.g., diet, exercise, oral medications, and insulin) as inputs and provides the resulting patient state $X_{t+\Delta}$ at a future point in time $t + \Delta$ (Dutta et al. 2005; McCabe et al. 2010). The state vector X_t includes, for example, patient characteristics (blood glucose level, systolic blood pressure, etc.) and current doses of medication. X_t can be decomposed into controlled, manipulated, and disturbance variables (c_t, m_t, d_t). For example, this decomposition will assign blood glucose and blood pressure to the set of controlled variables, medication doses to manipulated variables, and patient age to disturbance variables.

A specific treatment regimen has been translated into an algorithm to yield the initial strategy s_0. This strategy defines what actions are applied to a patient at a given simulated visit. Using the patient model and initial strategy, we simulate one year of treatment for 10,000 diabetes patients, which we randomly select from the population described in 3.3. **run-process**$(\mathbb{I}, T, n)$ can now be run, setting $\mathbb{I}$ to contain the patients as instances for the process, T=1 year, and n=0 to use the initial strategy. The result of this step is a history set $\mathbb{H}$ containing the state of every patient at every visit.

In summary, this step is responsible for collecting the data about the performance of a given decision strategy, which is then analyzed and modified in the subsequent steps, as described below. Consistent with the opaqueness of dynamic problems, Procedo is not aware of the inner workings of the Simcare patient model or the treatment regimen. It only observes the actions taken by the treatment strategy and the resulting health outcomes.

5.3.2. Step 2: Identify Preconditions to Failure

Each recorded state in the history set $\mathbb{H}$ is evaluated according to a given metric (discussed below) by the predicate **is-failure** which returns True if a state is the result of a control failure. Failure is assessed based on a metric derived from the state variables. The question at the core of Procedo is: What information at a given point in time predicts that a control failure occurs in the future? After recording the results of **run-process** and identifying control failures, data mining is employed to investigate this problem. Specifically, a data mining problem is set up to find a set of conditions $\mathbb{F}$ that have predictive value for future failures. Procedo generates $\mathbb{F}$ by applying a data mining algorithm, which is represented by **identify-failure-conditions**($\mathbb{H}$) in

Algorithm 3, to the recorded history $\mathbb{H}$. Out of the set of conditions $\mathbb{F}$, one condition will be selected as the predicate **predict-failure** (as will be discussed in Section 5.3.3). A high-level description of **identify-failure-conditions** is provided in Figure 15.

Algorithm 3 improve-control

Input $\mathbb{I}$: a set of process instances providing information baed on which a strategy is improved
Input T: amount of runtime after every iteration of improvement
The termination conditions are described in the narrative.

1: $n \leftarrow 0$
2: **loop**
3: $\mathbb{H}$ = **run-strategy**($\mathbb{I}$,T,n) // Step 1: run process for T time for instances in set $\mathbb{I}$
4: $\mathbb{F}$ = **identify-failure-conditions**($\mathbb{H}$) // Step 2: identify conditions predictive of failures
5: f_n = **select-failure-condition**($\mathbb{F}$) // Step 3: select condition for improvement
6: m_n = **create-modification**($\cdot$) // Step 4: modify decision strategy
7: $s_{n+1} = \langle s_n, f_n, m_n \rangle$
8: $n \leftarrow n + 1$
9: **end loop** // Step 5: iterate

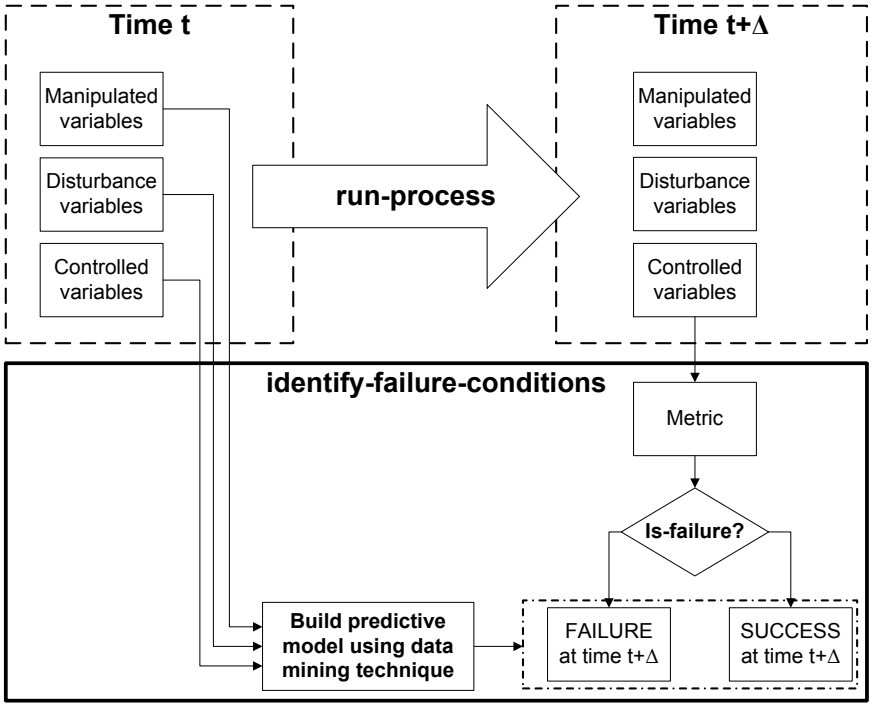

Figure 15. Overview of identify-failure-conditions

In the diabetes example, we ask two questions: (1) "Is there anything in the initial patient state (at time t=0) that predicts that we fail to bring the patient to evidence-goal within one year?" and (2) "Is there anything in the initial patient state (at time t=0) that predicts that we treat too aggressively and bring the patient to evidence-based goals too early?". The second question is important for two reasons: medical safety and cost-effectiveness. A medical safety concern was raised by a recent clinical trial, which showed that very aggressive treatment can increase fatality in patients (ACCORD Study Group 2008). The cost-effectiveness concern is that excessive resources are used to bring patients to evidence-based goals when less intensive treatment would be able to do the same. **is-failure** evaluates to True when a patient does not reach evidence-based goals after one year (we label this error *E*-) or reached goal in less than half the required time to goal (labeled error *E*+).

In order to identify the particular conditions under which a decision strategy fails, a data mining technique with *interpretable* results is used for **identify-failure-conditions**(ℍ). A decision tree algorithm, specifically the C4.5 algorithm, was chosen for this purpose (Quinlan 1993). (If prediction of failures without interpretability is acceptable, other techniques such as neural networks may also be used.) Decision tree induction uncovers informative patterns relating patient state to future control failures. Each node in the decision tree represents components of system state that predict with a certain level of accuracy, as discussed below, whether a strategy will succeed or fail.

Once the tree is generated, Boolean conditions can then be constructed by following the tree from the root to the leaves and conjoining each intermediate node with a logical AND (Quinlan 1987). For the diabetes care example, the C4.5 algorithm was used to build the decision trees based on 25 distinct patient state variables, including demographics, measurements of patient state, and medication dosages (McCabe 2012). The decision tree models were built and evaluated using ten-fold cross-validation. A fragment of a generated decision tree is shown in Figure 16; in our experiments, the full decision trees typically contained between 15 and 40 nodes.

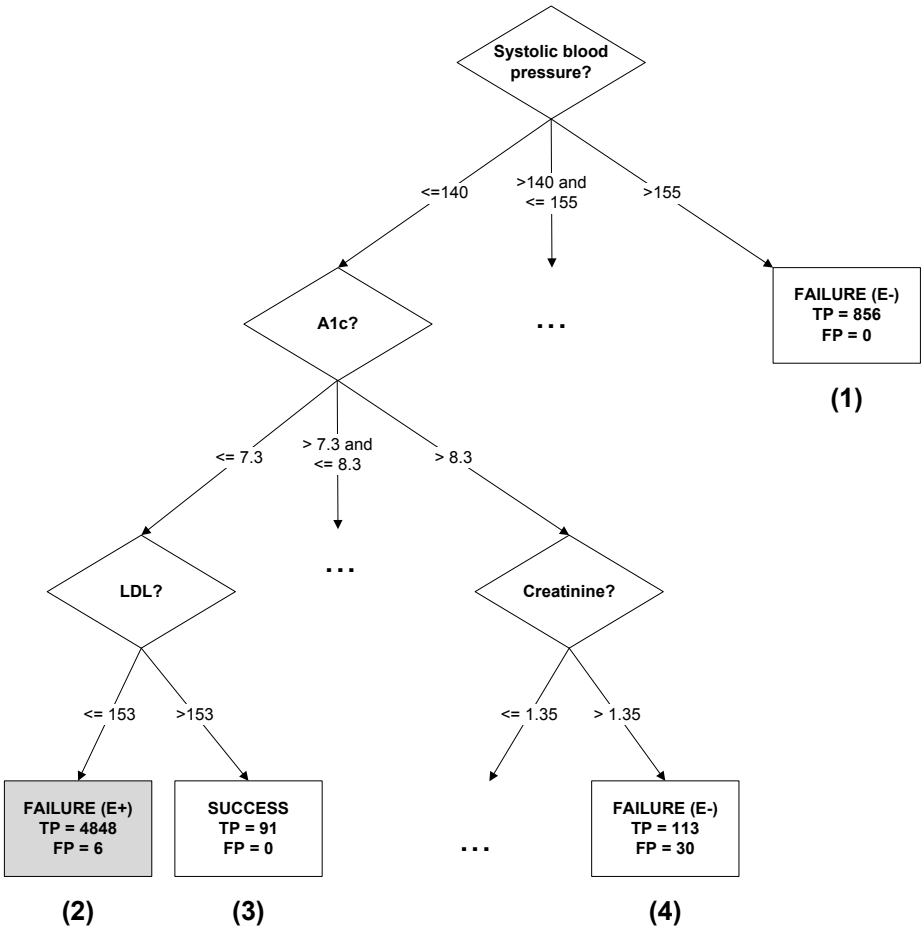

Figure 16. Example decision tree (TP = true positives; FP = false positives)

In the tree in Figure 16, three conditions predict control failures for the initial strategy: Node (1) predicts failures for all patients with systolic blood pressure (SBP) of greater than 155mmHg; Node (2) for all patients with SBP less than or equal to 140mmHg, A1c ≤ 7.3% and LDL ≤ 153mg/dl; and Node (4) for all patients with SBP ≤ 140mmHg, A1c > 8.3% and serum creatinine > 1.35mg/dl (node (4)). Patients in nodes (1) and (4) are not brought to goal within one year of treatment (failures of type *E*-) while patients in node (2) are treated too aggressively (*E*+ failures). On the other hand, it is predicted that the initial strategy is successful in bringing patients with SBP ≤ 140mmHg, A1c ≤ 7.3% and LDL > 153mg/dl (node (3)) to goal in the desired timeframe.

The conditions are constructed by following the branches from the root of the tree down to the node that predicts either success, or failure of type *E*- (insufficient action), or failure of type *E*+ (excessive action). The goal of Procedo is to identify control failures, so the success nodes are disregarded. The entire tree is evaluated, and all conditions leading to failure nodes, including, in this example, nodes (1), (2), and (4) are added to a set of failure conditions $\mathbb{F}$. This step of generating the set of all conditions predictive of failures $\mathbb{F}$ is represented by **identify-failure-conditions** in Procedo (Algorithm **improve-control**). As mentioned above, a single condition for which the strategy will be improved is then chosen from the resulting set $\mathbb{F}$ in the following step.

5.3.3. Step 3: Select Conditions for Improvement

After enumerating all the conditions contained in the decision tree, one condition for which the decision strategy will be improved must be selected from $\mathbb{F}$ as the **predict-failure**$_n$ predicate:

$$\textbf{predict-failure}: C \times M \times D \rightarrow \{true, false\}$$

In order to make meaningful improvements, it is desirable that the selected condition is of a certain quality, for example, in terms of its predictive accuracy and support (Tan et al. 2006). Intuitively, to gain the most from adjusting the strategy, this condition should result in a large number of true positives Q^+, that is, correctly identified failures, and a low number of false positives Q^-.

The total number of instances for which **predict-failure**$_n$ evaluates to True is therefore given by $Q^+ + Q^-$. Two common metrics to evaluate decision tree rules are accuracy and support. Accuracy represents the *positive predictive value*, i.e., the proportion of instances captured by the candidate condition that are in fact true positives, i.e., $\frac{Q^+}{Q^+ + Q^-}$.

The support of a candidate condition is the absolute number of instances it contains, $Q^+ + Q^-$. A minimum threshold for support is desirable to avoid the selection of a condition covering a small number of cases that is due to an artifact of a particular dataset rather than an underlying pattern in the process. If a rule accurately predicts a failure because it only applies to a single patient, it is questionable that it would generalize to a

larger population (Forster 2000). If more sophisticated selection criteria are desired, costs for true and false positives can be incorporated, in essence making the decision-tree cost-sensitive, for example, as discussed by Zhao (2007). Once the selection metrics have been decided, an algorithm **select-failure-condition** can be implemented to choose a single rule from $\mathbb{F}$ in accordance with these metrics. Below we give a formal definition of our selection Algorithm **select-failure-condition**.

Algorithm 4 select-failure-condition

Input $\mathbb{F}$: candidates to select from, i.e., a set of conditions predictive of control failures
Input $\mathbb{I}$: a set of process instances used to evaluate accuracy and support of conditions in $\mathbb{F}$
Input *minsupp*: Minimum required support; conditions that are not supported by at least this proportion of instances will not be considered.
Input *minacc*: Minimum required accuracy; conditions that cannot meet this threshold will not be considered
Output: the **predict-failure**$_n$ predicate for iteration n of Procedo

1: **for** each candidate condition f in $\mathbb{F}$ **do**
2: Calculate the number of true positives Q+ and false positives Q- for f
3: $f.accuracy = \frac{Q^+}{Q^+ + Q^-}$
4: $f.support = Q^+ + Q^-$
5: **if** $f.support < (minsupp \times |\mathbb{I}|)$ **then**
6: remove f from $\mathbb{F}$
7: **end if**
8: **end for**
9: **if** $|\mathbb{F}| > 0$ **then**
10: sort $\mathbb{F}$ by $f.accuracy$ in descending order, i.e., $\mathbb{F} = \langle f_0, f_1, \ldots \rangle$
11: **if** $f_0.accuracy > minacc$ **then2**
12: **return** f_0
13: **else**
14: Procedo terminates
15: **end if**
16: **else**
17: Procedo terminates
18: **end if**

To avoid finding spurious predictive conditions in the diabetes example, we apply Algorithm **select-failure-condition** and define a minimum threshold of 1% of the model-building data as support, i.e., *minsupp* = .01. Among the failure-predicting decision tree rules that satisfy the *minsupp* threshold, the one with the highest accuracy is chosen. We terminate the search for improvements if no more rules with an accuracy of at least 90% can be found, i.e., *minacc* = .9. The focus on accuracy stems from the critical nature of the patient care context; more forgiving domain contexts may put a focus on how many instances could be positively affected overall (expected number of true positives) or use a cost-sensitive approach.

The selection measures for the example tree shown in Figure 16 are listed in Table 12. First, all failure conditions with a support of less than 1% are removed. The remaining conditions are sorted by accuracy in descending order and the first condition in the resulting set is chosen. Therefore, the selected condition is "SBP > 155mmHg" and now represents **predict-failure** for the improved strategy. (The selection process described here, based on accuracy and support, can be simplified when using decision tree classifiers by configuring the model to require the desired level of minimum support and to stop splitting when dropping below a specified level of accuracy. This simplifies the implementation of Procedo while conforming to what has been described in this section conceptually.)

Table 12. Selection criteria for failure conditions in the example decision tree (Figure 16)

Condition	True Positives	False Positives	Support	Accuracy
(1) SBP > 155mmHg	856	0	856	1.0000
(3) SBP ≤ 140mmhg, A1c ≤ 7.3%, LDL ≤ 153mg/dl	4848	6	4854	.9988
(4) SBP ≤ 140mmhg, A1c > 8.3%, Creat ≤ 1.35mg/dl	113	30	143	.7902

5.3.4. Step 4: Improve Decision Strategy

Step 3 identified that, for the condition (or conditions) described by in **predict-failure**$_n$, s_n needs to be modified. Identifying this modification is the goal of **create-modification** (Step 4 in Algorithm **improve-control**). A modification is a set of rules to change the values of the manipulator variables m_t in a manner that will affect the system to be controlled more or less intensely than the previous strategy s_{n-1}.

The identification of an appropriate action in response to the selected failure condition depends on the type of failure (E- or E+) and the specifics of the application domain. Procedo is domain-independent and, therefore, permits each domain to apply its appropriate methods and domain expertise to identify and select improvements. This enables decision makers to avail themselves of methods including, but not limited to, means-ends analysis, expert deliberation, documented guidelines, choosing from formally specified set of candidate improvement actions, or simulation of possible alternatives to choose the most promising one. For the purposes of this paper, we use the simulation of alternatives selected by means-ends analysis and vetted for medical safety.

The goal in this step is to select the most effective modification that will eliminate the control failure. The type and magnitude of the possible adjustment is domain-specific and can be determined by different metrics, such as cost. The decision strategy is augmented to eliminate failures of type *E*- or attenuated to eliminate failures of type *E*+. In diabetes care, intensification of a treatment strategy can be achieved in three major ways: (a) by treating more than one condition (e.g., blood glucose, blood pressure, or cholesterol) at a given visit; (b) by scheduling more frequent visits in order to have more opportunities to treat; or (c) by intensifying treatment itself, for example, by administering a combination therapy of an oral agent and insulin in accordance with clinical guidelines (Mazze et al. 2005). Conversely, a treatment strategy can be attenuated by (a) treating fewer conditions at a given visit (if more than one is treated); (b) scheduling less frequent visits; or (c) using weaker treatments (single instead of combination medications, or smaller doses).

Table 13. Simulation results for three improvement candidates

Modification	Proportion at goal	Treatment cost (mil. $)	Event rate
(0) Without modification	.7778	5.82	.222
(a) Treat BP and other condition at same visit	.8282	6.12	.220
(b) Schedule additional visit for BP treatment	.8002	6.29	.221
(c) Stronger initial BP treatment	.8099	5.84	.221

In Table 13, we show the performance of three candidates for modifications for failure condition (1) from the decision tree in Figure 16, which was chosen for modification in Step 3: (a) treat blood pressure at the same time as another condition, (b) schedule an additional visit to treat blood pressure, and (c) intensify treatment, i.e., start

with a greater dose of blood pressure medication. The treatment costs for each option were calculated following the methodology outlined in Section 3.5. There is a small difference in the event rates for each improvement, making it not readily apparent which choice is the most cost-effective. This illustrates the complexity of the example problem. Is the difference between an event rate of .220 for candidate (a) and .221 for candidate (c) worth the $280,000 greater cost of (a) compared to (c)?

Using the cost-effectiveness approach discussed in Section 2.3.3 (and illustrated in Figure 2), each candidate is evaluated in terms of cost-effectiveness and plotted in Figure 17. Any two points on the same grey dotted line have the same value for the cost-effectiveness function Φ. Even though, at first glance, it may seem that (c) is most cost-effective because it has a very steep drop, in reality, (a) performs better, i.e., lies on an isobar with a lower intercept than (c) and has the lowest value of Φ out of the three candidates. Consequently, (a) selected as the improvement to create the next iteration of the decision strategy.

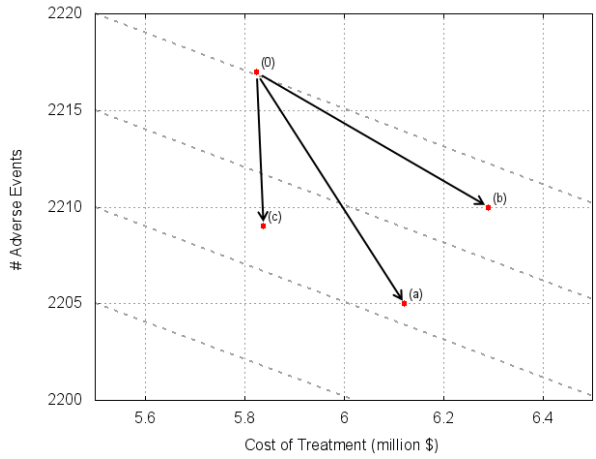

Figure 17. Comparison of three candidates for strategy improvement (presented in Table 13)

As described in Algorithm **control-strategy**, the improved strategy s_{n+1} is completely defined by three items: the current strategy s_n, the predictor for future control failures **predict-failure**$_n$, determined in Step 3, and **modify-action**$_n$ created in this step, that is applied when a failure is predicted. Therefore, s_{n+1} is now the resulting improved strategy (version $n + 1$).

5.3.5. Step 5: Iterate

After making an improvement to strategy s_n, the process (i.e., Steps 1-4) is run again, this time governed by the improved strategy s_{n+1}. The states can again be recorded, quality outcomes assessed, conditions for additional improvement identified using data mining, and further improvements to the strategy can be made. If a point is reached at which no more actionable conditions can be discovered, or the set of possible modifications has been exhausted, then Procedo has identified boundary conditions for

this particular dynamic decision making problem. For example, it may be possible that, even with all the available intensification options, a patient with a very high initial blood pressure cannot reach the evidence-based goal within one year of treatment.

In summary, we have proposed an approach to improve decision strategies that are employed in dynamic environments. While the initial strategies may be pure feedback strategies, each iteration of Procedo introduces feedforward anticipation to the process by applying data mining to examine preconditions to control failures (Brehmer 1990; Seborg et al. 2004). In other words, the approach identifies subgroups - in our example, subgroups of patients - for which the original strategy failed to achieve desirable outcomes, either by taking insufficient or excessive actions. For these subgroups, and only these subgroups, we propose the identification of a minimal modification that will eliminate the control failure.

Below, we demonstrate this iterative improvement process and present results from applying it to the example of diabetes care using three different initial treatment strategies.

5.4. Validation of the Proposed Approach: Procedo-Modified Diabetes Care

To demonstrate that Procedo improves decision strategies by making appropriate modifications, augmenting or attenuating when necessary, three scenarios were created with different starting points, i.e., three very different initial strategies. The first scenario uses a very weak initial strategy referred to as *MIN*, informed by "usual care" literature,

in need of augmentation (Phillips et al. 2001; Selby et al. 2003; B. R. Shah et al. 2005). One can think of this strategy as very frugal and assuming that at least some patients may get better even without costly treatment. In stark contrast, the strategy in scenario 2, referred to as *MAX*, is extremely aggressive (and consequently costly), informed by diabetes specialist guidelines (Mazze et al. 2007). *MAX* can be thought of using any means necessary to treat patients regardless of cost. The third scenario lies between the two extremes *MIN* and *MAX* and follows a strategy *EXP* by observations of experts in clinical practice (Ramsey 2010).

In each scenario, we are interested in the number of patients that can be brought to evidence-based goals within one year of treatment. We evaluate the overall outcomes for each iteration in terms of (a) number of patients at goal, (b) risk resulting from not being at goal, and (c) cost of treatment. The intent of bringing patients to evidence-based goals is to reduce their risk of adverse clinical events (e.g., heart attacks, strokes, kidney failure, etc.). Therefore, we quantify the risk resulting from not being at goal by calculating the risk of such an event in a ten-year horizon using the UKPDS Risk Engine (Stevens et al. 2001) and applying Monte Carlo simulation to translate the risk estimate into events for the simulated patients (as discussed in Appendix A). Using cost estimates for adverse events and cost-effectiveness thresholds, we compute the cost-effectiveness Φ (see Section 2.3.3) for each treatment strategy. (For details about cost estimation, please refer to Section 3.4). The cost-effectiveness function Φ encodes the relevant domain-knowledge to assess whether a strategy X performs better, worse, or the same as strategy Y in terms of cost-effectiveness (Bell et al. 2006; O'Brien et al. 2003; Chapman

et al. 2000; O'Brien et al. 1998). In the graphs presented below, we will show a level of equal performance as a grey dotted line representing an isobar, i.e., each strategy on the same line is equally cost-effective. Better-performing strategies reside on isobars with intercepts lower on the x-axis.

All results presented in the tables below are after one year of treatment. The conditions identified for **predict-failure** in Step 3 specify a subgroup of the patients for which the treatment strategy fails. In scenario 1, the treatment strategy under-performs, in scenario 2, the strategy tends to treat excessively, and in scenario 3, the strategy is insufficient for some patients but may be excessive for others.

The proportion of *patients at goal* represents how many patients reached evidence-based goals for all three conditions, blood sugar, blood pressure, and cholesterol. The number of visits is the average number of patient/physician interactions for the one year timeframe. The number of *glucose moves* represents how many times a physician prescribed a blood sugar medication or increased the dose; *blood pressure moves* and *cholesterol moves* are analogous. The number of moves represents the total for the entire treatment period of one year and for all 10,000 patients. In Study 1, the proportion of moves was presented to demonstrate how different prioritization strategies focused on different conditions. In this study, to focus attention not only on the mix of moves but also on treatment intensity, the absolute number of moves is displayed.

The *cost for treatment* for all patients was calculated using estimates for the cost of office visits and medications provided by a regional healthcare system. The cost of

adverse events to determine the cost-effectiveness isobars is derived from the UKPDS Outcomes Model (Clarke et al. 2004). (The cost estimation methods are discussed in Section 3.4.) *Event rate* represents the proportion of patients that experienced heart attacks or strokes in the ten-year period following the treatment and is estimated using the UKPDS Risk Engine (Stevens et al. 2001).

5.4.1. Augmenting *MIN*

Table 14 shows the results for augmenting a very weak but inexpensive strategy referred to as *MIN*. In the first iteration, Procedo identified that *MIN* did not bring patients with initial systolic blood pressure (SBP) greater than 135 mmHg to goal within one year (*E*-), i.e., **predict-failure**$_1$ evaluated whether "initial SBP > 135 mmHg" was true for a given patient. Following the proposed approach, an improvement was selected for **modify-action**$_1$, in this case, to schedule an additional visit for patients for which **predict-failure**$_1$ was true. Adding this rule to strategy s_0 produces the improved strategy s_1, the final product of iteration 1. For the second iteration, Procedo predicted that patients who have an initial A1c greater than 7.4%, now treated with s_1, still do not reach evidence-based goals for all three conditions (*E*-). Again following the **create-modification** approach, the augmentation of scheduling an additional visit was selected for s_2, increasing the number of treatment opportunities for the simulated physicians.

Table 14. Augmenting *MIN*

Iteration	Failure type	Avg. # visits per patient per year	# glucose moves	# BP moves	# lipids moves	Proportion at goal	Cost of treatment ($ mil)	Event rate
0		1.00	3358	3430	4156	.40	3.4	.237
1	***E-***	1.35	4253	6699	5620	.46	3.8	.233
2	***E-***	1.73	6866	8273	7420	.52	4.4	.230
3	***E-***	2.00	7678	10848	8479	.57	4.7	.227
4	***E-***	2.18	8333	12671	9306	.61	5.0	.226
5	***E-***	2.51	10260	14819	10882	.67	5.5	.223
6	***E-***	2.74	11837	16551	11914	.71	5.8	.222

As one would expect, given that *MIN* is a very weak strategy, each iteration requires intensification of treatment for a subgroup of patients. Thus, every iteration increases the cost of treatment and the number of patients at goal and decreases the number of adverse events (lowers the event rate). Compared to the baseline strategy (*MIN*), in iteration #6, the Procedo-augmented strategy brings 3,150 more patients to goal for an additional $2.4 million and prevents 156 adverse events. The trade-offs between adverse events and treatment cost for *MIN* (as well as *MAX* and *EXP*) and the strategies resulting from each iteration of Procedo are represented graphically in Figure 18. It can be seen that each iteration is more cost-effective than the previous, as measured by the cost-effectiveness function Φ described above.

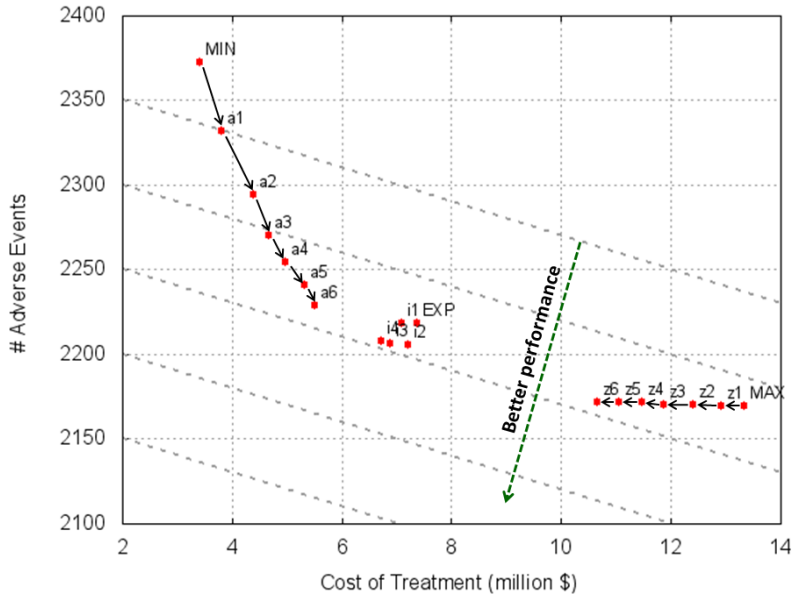

Figure 18. Effect of Procedo-modification of *MIN*, *MAX*, and *EXP* on treatment cost vs. adverse events trade-off

5.4.2. Scenario 2: Attenuating *MAX*

Table 15 shows the results for weakening an extremely intensive strategy that we refer to as *MAX*. In the first iteration, Procedo identifies that treatment is too intensive for the subgroup of patients with an initial A1c ≤ 7.4%, initial LDL ≤ 170mg/dl, and initial SBP ≤ 175mmHg. For this subgroup, the selected attenuation is to reduce the visit frequency. In the next iteration, Procedo identified the subgroup of patients with initial SBP ≤ 163mmHg and A1c ≤ 8.5%, for which treatment intensity can be reduced further by not treating both blood pressure and blood glucose at the same time.

In the previous scenario, improving the *MIN* strategy, treatment actions had to be added for every iteration, resulting in higher costs. In this scenario, since *MAX* typically

uses all available treatment actions, the only choice available to Procedo is to remove treatment actions at every iteration, thus saving costs. However, in order for the resulting strategies to be cost-effective, these cost savings cannot be offset by significant decreases in the number of patients at goal or significant increases in the number of adverse events. (Otherwise the cost of complications from adverse events would cancel out the savings in treatment costs.) The performance for this scenario is documented in Table 15.

Table 15. Attenuating *MAX*

Iteration	Failure type	Avg. # visits per patient per year	Glucose moves	BP moves	Lipids moves	Proportion at goal	Cost of treatment ($ mil)	Event rate
0		12.00	24306	27335	49327	.92	13.3	.217
1	***E+***	11.33	24115	26944	46782	.92	12.9	.217
2	***E+***	10.51	23352	26389	43584	.92	12.4	.217
3	***E+***	9.71	22310	26005	40479	.92	11.9	.217
4	***E+***	9.07	21941	26037	38014	.92	11.5	.217
5	***E+***	8.42	21742	25539	35537	.92	11.0	.217
6	***E+***	7.79	21540	25132	33125	.92	10.6	.217

After six iterations, the Procedo-attenuated strategy reduced the number of patients at goal by only 40, but saved $2.7 million in treatment cost without negatively impacting the event rate. Using the weakest strategy (MIN) would save an additional $7.2 million in treatment costs compared to iteration #6, but results in 201 additional adverse events and reduces the number of patients reaching goal by 57%. The trade-offs are represented in Figure 18. The graph confirms that each iteration is more cost-effective than the previous one.

It was demonstrated that Procedo can be used to augment as well as attenuate treatment strategies by applying it to minimally (weak) and maximally intensive (strong) initial strategies. Each iteration resulted in a more cost-effective strategy than the previous iterations. For the weak strategy, Procedo increased cost-effectiveness primarily by improving outcomes, i.e., bringing more patients to goal and lowering the event rate. For the strong strategy, Procedo increased cost-effectiveness primarily by saving cost without negatively impacting outcomes, i.e., the proportion at goal and event rate remained the same for all six iterations in Table 15. To complete the evaluation of Procedo, below, Procedo is applied to a strategy observed in practice that can be improved by augmenting for some patients and attenuating for others.

5.4.3. Scenario 3: Improving *EXP*

In this scenario, Procedo is used to improve a treatment strategy informed by observations of diabetes experts treating patients in clinical practice (Ramsey et al. 2010). This is underlined by the fact that even the initial strategy *EXP* is more cost-effective than *MIN* and *MAX*. The performance of the strategies resulting from the application of Procedo are shown in Table 16 and graphically represented in Figure 18 and Figure 19. In order to demonstrate the differences between iterations clearly, Figure 19 uses a finer scale than Figure 18. The thicker, dotted line in Figure 19 represents the performance level of *MAX*.

Table 16. Improving *EXP*

Iteration	Failure type	Avg. # visits per patient per year	Proportion at goal	Cost of treatment ($ mil)	Event rate
0		6.00	.80	7.36	.222
1	*E+*	5.51	.80	7.08	.222
2	*E-*	5.60	.82	7.20	.221
3	*E+*	5.12	.81	6.88	.221
4	*E+*	4.80	.81	6.70	.221

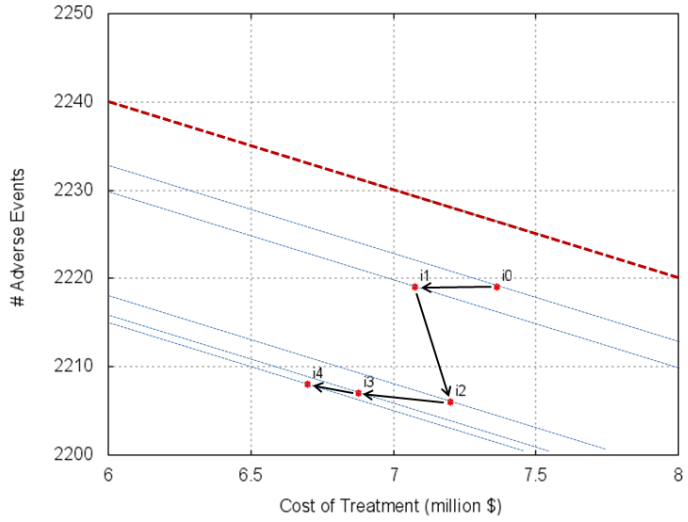

Figure 19. Effect of Procedo-modification on *EXP* on treatment cost vs. adverse events trade-off

For the first iteration, Procedo has identified a subgroup for which attenuation was the appropriate response, resulting in fewer treatments and, thus, cost savings. In the next iteration, the selected subgroup required augmentation and resulted in fewer adverse events but an increase in treatment cost. The following two iterations each saved costs by further attenuation. Each iteration produced a strategy performing better in terms of cost-effectiveness, as demonstrated by it residing on an isobar with a lower intercept. In this

particular example, the superior performance of iteration 4 is also evident in the results in Table 16, which show that the iteration 4 strategy is both less expensive and has fewer adverse events than the initial strategy (iteration 0).

As the results show, for all three strategies, each iteration of the developed approach results in a strategy that either lowers the event rate or the cost of treatment compared to the previous version, resulting in more efficacious or cost-effective treatment strategies.

5.5. Discussion and Conclusions

In this study, a principled approach named Procedo was developed to improve strategies which guide decision making in dynamic contexts. Procedo uses an inductive machine learning algorithm to identify areas for improvement and provides guidance in the selection of iterative improvements so that the resulting strategy is more cost-effective, either by only consuming as little additional resources as necessary to achieve the desired goal, or by reducing the use of resources when a strategy applies excessive actions. Procedo can be applied to dynamic contexts as long as the system to be controlled has observable variables and opportunities to augment or attenuate the strategy. Observability is required to obtain feedback for the strategy to be improved. To show the efficacy of Procedo, it use was demonstrated its use in the context of chronic disease management, namely, for the treatment of type 2 diabetes mellitus patients. Procedo was shown to be effective in strengthening weak and attenuating strong treatment strategies, including a strategy observed in clinical practice. In each case, Procedo has increased the cost-effectiveness of the strategy. Starting from very different

performance levels, all three initial strategies were brought to a similar, higher level of cost-effectiveness (see Figure 18). These results support that Procedo is a viable method to identify specific conditions for improvement and then iteratively fine-tuning the improvements in order to make cost-effective use of resources.

In addition to its effectiveness in improving a broad range of initial strategies, Procedo can be customized in a number of ways to match the dynamic context to which it is applied, such as by choice of different evaluation metrics (i.e., cost-effectiveness trade-offs, as discussed earlier), different predictive techniques to identify failure conditions, different selection criteria for failure conditions, and different methods for modifying the strategy.

In the main example, Procedo was shown to identify subgroups of patients for which treatment strategies needed to be modified. In this case, the selected quality metric assessed the overall treatment strategy, which did not differ between visits. In the same context, the proposed approach can be applied to a problem of finer granularity: improving the treatment patients receive at specific visits. Using an analogy to manufacturing, these two problems can be regarded as reducing the overall defect rate of the final good vs. reducing errors during different phases of manufacturing. Rather than evaluating whether patients are at evidence-based goals at a given point in time, this formulation considers whether patients make at least an expected minimum progress toward goal but, at the same time, do not make dangerously excessive progress. The

range of acceptable progress for this formulation can be derived from randomized clinical trials and medication effect curves from drug studies.

It may be possible to make the approach even more powerful by using alternative data mining techniques, such as neural networks, support vector machines, or Bayesian classifiers, or more fine-tuned decision trees than the C4.5 decision trees that were selected in the example for their interpretability. In addition, in special cases, when all the available adjustments to a strategy can be formally specified and enumerated, it may be possible to formulate the approach in a way that an improvement is automatically selected and, therefore, a fully automated version of Procedo could be implemented (once domain-specific quality and evaluation metrics have been defined). The use of alternative data mining techniques and full automation constitute interesting and important directions for future work.

Procedo is not limited to the medical domain but rather constitutes an approach with broad applicability. For example, Procedo could be applied to an emergency management problem, such as fire-fighting. In this case, the control variables are characteristics of the fire (for example, coordinates to which it has spread, temperature, etc.); the manipulated variables could be methods of extinguishing, such as water and foam, and orders to send fire fighter to specific locations; and disturbance variables are wind speed and direction. The goal is defined as the complete extinction of the fire; a point-in-time metric to assess success or failure could be the size of the area covered by fire at a given time; a progress metric would be a measure of whether the fire expands rather than contracts (change in

size over time). A conceivable outcome is that Procedo determines that a certain method of fire-fighting is inappropriate for a certain type of fire, or location, or type of building, or that a specific unit of fire-fighters always fails to fight fires that meet certain initial conditions.

Traffic control at airports represents another possible domain in which Procedo could provide insights leading to improvement. Here, a general goal may be to reduce touchdown-to-gate time, however, during emergencies, other goals override. A controlled variable is arrival at gate time, a manipulated variable is the gate assignment for a plane, and examples of disturbance variables are weather conditions, flight diversions, etc. Procedo may identify sub-groups of flights that tend to cause delays before arriving at the gate, or show that there are consistently delays at certain intervals for which additional capacity would be necessary.

Procedo can be applied to any dynamic decision making context that features ways to manipulate the system and feedback related to the variables to be controlled. Procedo gains considerable power and is applicable in many domains due to the fact that the underlying processes and initial strategy do not have to be formally specified or explicitly modeled. The incremental wrapper-based approach can yield improvements even when the initial strategy is a "black box", as long as observations of the process are available. Using the formal description provided in this paper, Procedo can be readily implemented in simulation environments to assess potential changes in decision strategies and support process design. As demonstrated in the diabetes care example, the proposed approach is

easily paired with simulation technology for the modeling of process control improvements. Benefits of the pairing with simulation include the consistent availability of process data (no missing data), the ability to analyze and improve upon counterfactual scenarios (e.g., "no additional treatment" in medicine) to discover new strategies, and rapid speed of studies (e.g., no time and resources needed for subject recruitment, compensation, and related activities). Of course, for simulation studies to be of value, the simulation must accurately represent the system to be improved; for example, as indicated by the extensive validation work on the Simcare diabetes patient model (McCabe 2012; McCabe et al. 2010).

In conclusion, Procedo is a novel data-driven approach to aid decision makers operating in environments that require series of decisions, in which prior decisions constrain future choices, and the environment changes on its own as well as in response to actions taken by the decision maker. It may be especially useful in environments that tend to be challenging for human decision makers, namely environments which feature delay of feedback, limited ability to examine the inner workings of a system, and multiple competing goals, such as treating chronic disease patients in a healthcare setting. In this particular context, Procedo has been demonstrated to produce significant cost-effective improvements to treatment strategies by capturing benefits from improved feedforward decision making while controlling costs by taking actions only where necessary. In a real-world healthcare application, the proposed approach could be used to enhance electronic medical records and clinical information systems by providing actionable analysis of the data residing in these systems.

6. Study 3: Evolving dynamic decision strategies for a given environment

In chronic disease care, the objective of a decision strategy is to bring patients to evidence-based goals. A given patient can be represented as a point in a state space (defined by demographics, health states, and specific risk factors, e.g., age, gender, A1c, SBP). An example of a three-dimensional patient state space is given in Figure 20, showing a patient's A1c, LDL, and risk of a cardiovascular event.

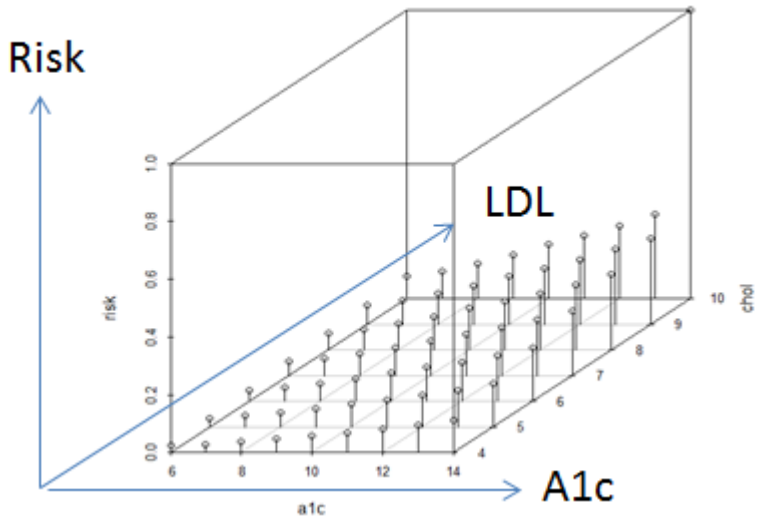

Figure 20. Example of a three-dimensional patient state space

A decision strategy applies treatment actions in order to bring a patient from one part of the space to another (e.g., more desirable) part. Strategies select actions based on the patient's state and the mental model employed by a given strategy. Consequently,

application of a strategy over time determines the path a patient takes through the state space, as shown in Figure 21.

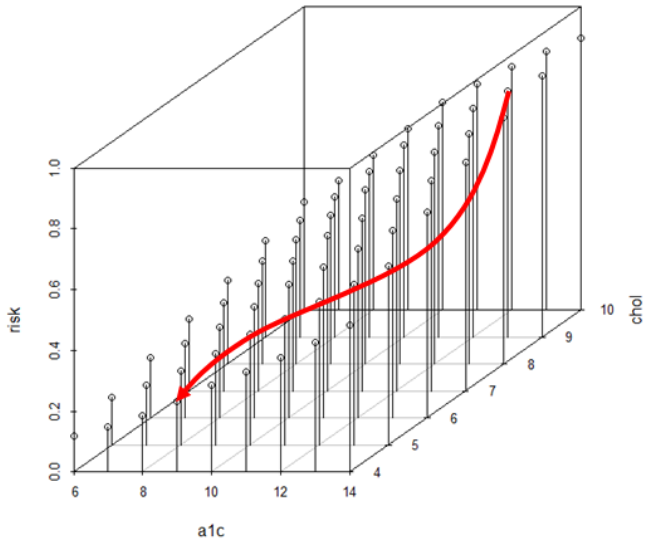

Figure 21. Example a treatment as a path through the patient state space

A given decision strategy prescribes paths (pathways of care) for every patient in the state space. Depending on the patient, these pathways may take different shapes. For example, Figure 22 shows treatment pathways for three different patients, one with high glucose and high lipids, one with low glucose and high lipids, and one with high glucose and low lipids. The goal for all three patients is the same, in the low glucose and low lipids area of the state space.

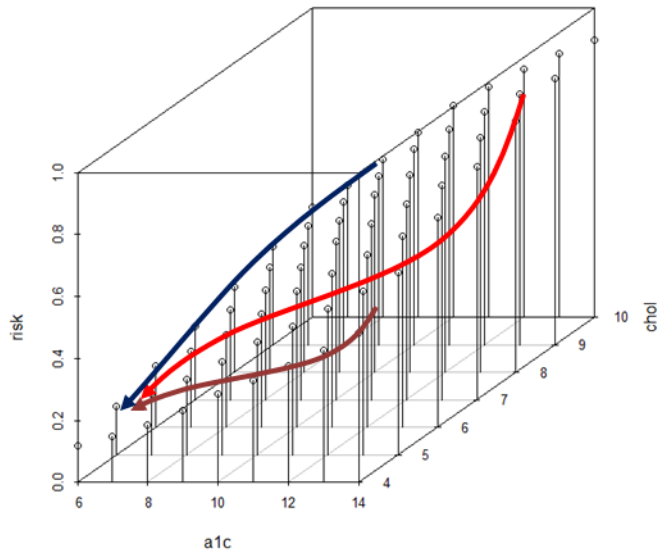

Figure 22. Treatment paths for three different patients produced by a strategy

Different pathways of care result in different cost/benefit tradeoffs. For example, some strategies may be less costly but are unable to produce pathways that bring certain types of patients to goal (for example, *MIN* from Study 2), while other strategies bring everyone to goal in a given timeframe but at a higher cost (e.g., *MAX* from Study 2).

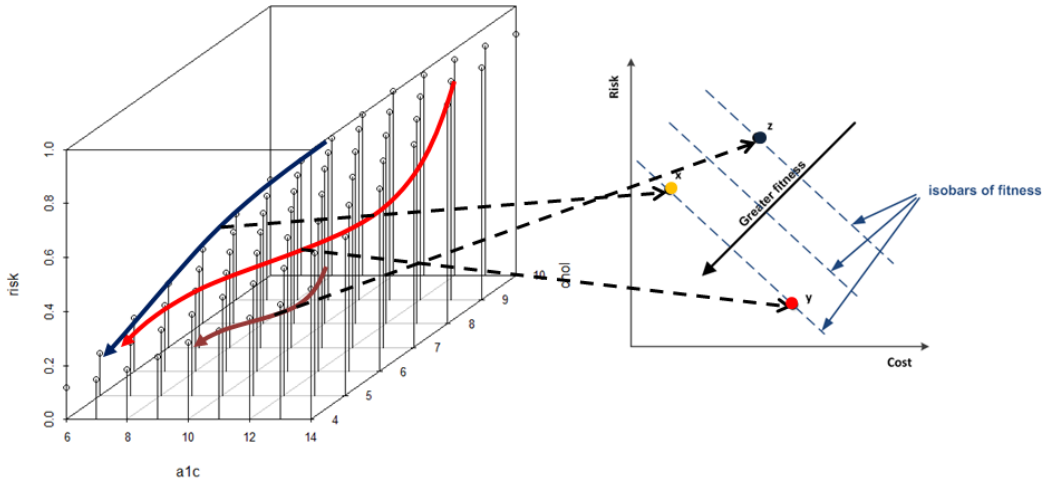

Figure 23. Treatment paths produced by a strategy and their resulting fitness

The costs and benefits of applying a given strategy can only be assessed once the strategy has been applied in a given environment. Study 1 investigated the costs and benefits of strategies based on principled approaches to clinical decision making, such as treating based on distance to goal or prioritizing by estimated risk reduction. Study 2 developed a specific approach to improving strategies with respect to costs and benefits. This approach kept the original strategy intact and amplified or attenuated an action selected by a given strategy when a failure was predicted. With each iteration of this approach, a new strategy was created that improved upon its predecessor with respect to a criterion, such as cost-effectiveness.

The study presented in this section assesses the costs and benefits resulting from application of a given strategy from a different perspective. Suppose a decision maker is asked to select a decision strategy from a set of choices. If a strategy y is known to clearly outperform another strategy z both in terms of cost as well as risk reduction, we

might expect the selection in favor of strategy y. Taking an evolutionary perspective, the two dimensions of costs and benefits determine the *fitness* of a strategy – its likelihood of selection. The cost-effectiveness framework from Figure 2 then becomes a fitness framework, as shown in Figure 24.

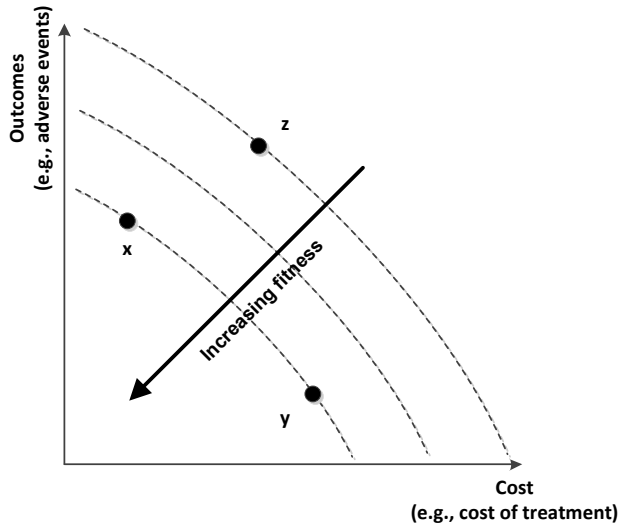

Figure 24. Cost-effectiveness framework reframed as a fitness framework

Current medical guidelines and treatments are subject to selection pressures through randomized clinical trials, cohort studies, meta-analyses etc. For example, a clinical trial for a given blood pressure drug may define fitness as the drug's ability to lower blood pressure by a statistically significantly greater amount than a placebo. The drug may also be found to cause higher rates of heart attacks and greater mortality (negative fitness) and be removed from the set of viable treatment options. A continuous evolutionary process updates treatments and guidelines. The study described in this section proposes a similar,

but explicitly formulated, process to evolve (generate) decision strategies for the treatment of patients with type 2 diabetes.

Our focus is to investigate potentially new strategies that adapt to specific problem environments through an evolutionary process. Particular questions of interest include: (1) Are such strategies substantially different from the ones examined in the previous studies? (2) Do they make use of different information or use the same information in different (e.g., better) ways?

These questions can be examined using a method from computer science referred to as *genetic programming*. The background of this method is outlined below.

6.1. Background on Genetic Programming

In this section, the term *program* is used to describe a set of instructions that can be executed by a computer. Later on, how decision strategies can be represented as programs is discussed.

In computer science, the traditional means of enabling a computer to solve a given problem is to write a program that specifically instructs the machine how to solve that problem. In artificial intelligence research, one question of interest was (and continues to be) whether computers can be made to solve problems without being given precise instructions on how to do so. Three key streams of research resulting from this question were the General Problem Solver, learning frameworks, and evolutionary methods.

The General Problem Solver was a computer program designed to solve any formalized symbolic problem, such as playing chess or proving a theorem (Newell et al. 1959). Its limitation was the need to explore a combinatorial number of possibilities to arrive at a solution, making most real-world problems intractable for investigation by means of the General Problem Solver (Newell & Ernst 1969). The approach to determine an optimal treatment strategy in Study 1 (within the strict constraints applied to that study) is modeled after the General Problem Solver approach by exhaustively searching the combinatorial space of potential treatments (i.e., a game tree).

The learning paradigm is based on the idea that a program should generalize from given examples, i.e., previously seen data, in order to improve future performance. A common definition is given here: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance on tasks in T, as measured by P, improves with experience E."* (T. M. Mitchell 1997)

Examples of machine learning include data mining algorithms, such as the decision tree classifiers used in Study 2. In Study 2, machine learning was used to identify conditions that occurred in observed (past) instances when a treatment strategy was about to commit an error. This knowledge was then used to predict future errors and compensate for them in advance. Using Mitchell's definition and substituting "computer program" for "strategy" is what was done in Study 2. Experience was provided in the form of patient history, the tasks were treatments, and performance was measured in terms of risk and cost.

Starting in the 1960s, a different type of answer to the question of how to make computers solve problems without specific instructions was suggested: let the computer evolve a program that solves the problem (Bäck 1996). This approach borrowed concepts from biological evolution and brought forth a class of methods referred to as *evolutionary algorithms*. Two of the most prominent methods in this class are *genetic algorithms* and *genetic programming*. The two approaches differ primarily in the way that a problem is represented, and consequently, the structure of the evolved solutions. Genetic algorithms encode candidate solutions as strings that are manipulated, while genetic programming evolves executable computer programs (Holland 1992; Koza 1992).

An evolutionary process begins with a population of individuals with between-individual variation that affects the likelihood of survival and reproduction in a given environment. This variation is expressed in individuals' genetic code, which can be thought of as the blueprint for the final organism. For example, some species of bacteria have evolved to survive and thrive in an environment such as an arsenic lake that would be toxic and hostile for most other organisms (Oremland et al. 2004). When individuals reproduce, a part of their genetic code is brought into the next generation, i.e., they beget individuals with a similar blueprint and, therefore, become organisms with similar characteristics. Reproduction allows for the combination or increase of traits that promote survival in the environment, and therefore, offspring may have greater *fitness*. On the other hand, since reproduction may include a random recombination of the parent's genetic material, some offspring may have lesser fitness and will likely be selected against (fail to reproduce). In this way, maladaptive traits are removed from the genetic

material of the population over time. In addition to reproduction, the genetic code is affected by mutations, small random changes to the genetic code, that introduce additional variation and novelty, allowing for the possibility of diversity that does not derive directly from an organism's predecessors.

In genetic programming, a computer program represents the genetic code for solving a problem in the agent's environment. Instead of providing the computer with a specific problem-solving approach in the form of a finished program, the computer is provided only with a way to evaluate the solutions its programs generate: a *fitness function*. A fitness function is designed to measure the quality of problem solutions. The computer then generates solutions (in the context of this research, decision strategies), measures their fitness (e.g., risk reduction). Fit solutions are then recombined into new solutions, which may or may not result in greater fitness. Solutions with poor fitness are selected against and will no longer reproduce into new solutions.

To evolve programs in genetic programming, a set of building blocks and a set of rules must be provided. The rules describe valid ways of assembling the building blocks, similar to a grammar describing what constitutes valid sentences in a language. The building blocks may be combined in an arbitrary fashion as long as they conform to these rules. This ensures that the evolved structure is an executable program. To begin, an initial population of programs is randomly assembled. Each program is evaluated using the fitness function and allowed to reproduce with a probability proportionate to the program's fitness. For reproduction, genetic operators – crossover and mutation – are

applied. Crossover combines parts of two existing programs into a new program; mutation makes random changes to a program. As a result, parts of the more promising programs (with greater fitness) are recombined into new programs. If parts from the parent programs make different valuable contributions to the solution, the new program may produce a solution with greater fitness. This process is repeated for every generation.

Koza demonstrated that genetic programming can be a new means of creating problem solutions that have been previously created by other means (Koza et al. 2003). A powerful example in his work is the use of genetic programming to evolve alternative circuit board designs that perform the same functions as designs that were created by engineers. At least eleven designs discovered by genetic programming infringed on existing patents, and another ten duplicated functionality of a patented design in an alternative way (Koza et al. 1999; Koza et al. 2003). In some cases, the evolved designs performed better than the human-created solutions (Koza et al. 2003). The objective of the presented Study is to do the same for strategies for treatment of type 2 diabetes patients.

In addition to circuit board design, genetic programming has been successfully applied to a wide range of problems. It showed early success and remains a research stream of interest in symbolic regression (Koza 1992; Cai et al. 2006; Gustafson et al. 2005; Keijzer 2004). A symbolic regression example that illustrates the power this method was the rediscovery of Kepler's Third Law using only the distances of the planets of our solar system to the sun and a basic set of mathematical functions (addition,

subtraction, multiplication, division, square root, sine, cosine) (Koza 1992). Solutions evolved through genetic programming have competed with and outperformed humans in the invention of database search algorithms (Spector et al. 1999), design of analog circuits (Koza et al. 1999; Koza et al. 2003), and in Mate-in-N chess problems (Hauptman & Sipper 2007). Further applications include visual and auditory processing (Zhang & Smart 2006; Xie et al. 2006; Usman et al. 2007), financial markets (Neely et al. 2009; Chen & Liao 2005; T. Yu et al. 2004), geological bomb fragment detection (Deschaine 2006), Internet game playing (Funes et al. 1998), and lossless data compression (Kattan & Poli 2008).

Below, it is explained how decision strategies can be represented as a particular type of computer program. Using this representation, genetic programming can be applied to develop programs that can be translated back into decision strategies with improved fitness, which may defined as success in bringing patients to goal, lowering risk, or improving cost-effectiveness.

6.2. Fitness Landscape

A decision strategy guides patients through a state space by means of actions, specifically, treatment and scheduling. Given a fitness function, such as a risk estimator, this state space can be defined as a *fitness landscape*. A fitness landscape assigns a fitness value to every point in the patient state space (Langdon & Poli 2002). A simplified example of two fitness landscapes is shown in Figure 25, in this case, defined by blood glucose (A1c) and cholesterol for female patients of age 40 and age 70. The actual fitness

landscape is a high-dimensional space with as many dimensions as there are state variables, plus the fitness dimension.

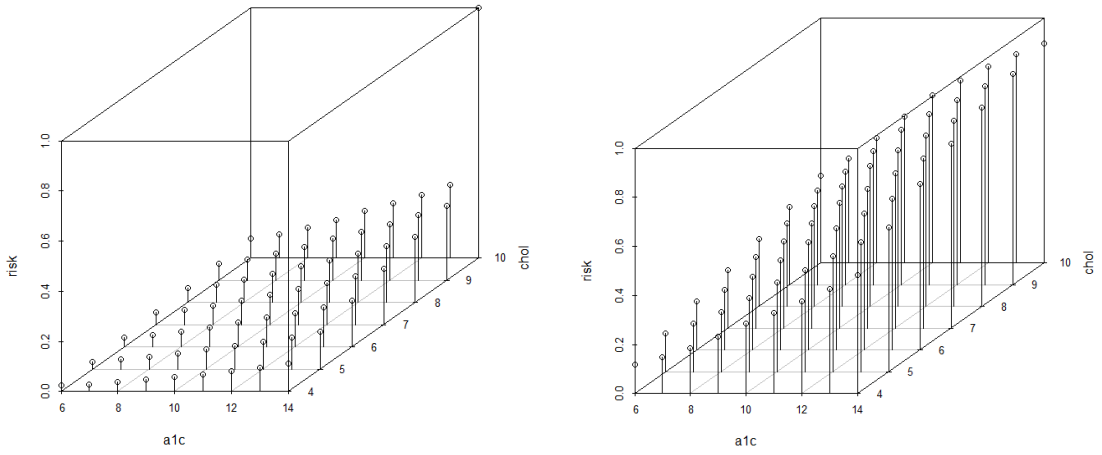

Figure 25. Example fitness landscapes with respect to A1c and total cholesterol for a patient with age 40 (left) and age 70 (right)

Each patient occupies a point in this fitness landscape. The objective of treatment is to bring patients to lower risk areas in the space. (The case of cost-effectiveness will be discussed later.) Given this objective, we are interested in strategies that reduce patients' risk. Instead of manually creating algorithms for these strategies (as was done in Study 1), or learning from experience to improve a strategy (Study 2), we are interested in evolving strategies that reduce risk. We propose that this is possible using genetic programming with a fitness function that increases when risk decreases. This approach will evolve strategies with greater fitness by adapting to the fitness landscape and making

appropriate moves, based on the current location in the landscape, to move to greater fitness areas.

Evolution is the result of variation and selection. Variation is generated the first time by means of initial seeding, i.e., randomly generating an initial set of strategies. Variation is thereafter created in every generation by reproduction and mutation, while selection is accomplished by making survival (and, consequently, reproduction) a function of fitness. Strategies that fail to lower risk will be selected against and become extinct while successful strategies will survive and pass their decision rules on to offspring strategies, which may achieve even higher levels of risk reduction. The following sections describe how a decision strategy is represented so that genetic programming can be used to evolve it, the specific operations that manipulate this representation, and finally the fitness functions used to evaluate the resulting decision strategies.

6.3. Representing Decision Strategies as Genetic Programs

Before decision strategies can be generated by means of an evolutionary process, it is necessary to create a representation that can be manipulated by this process. On a conceptual level, a decision strategy is represented by the process control model shown in Figure 26 (see also Section 1.2). Everything except for the shaded box, which represents the system to be controlled, is part of the decision strategy. The representation of a decision strategy must contain all the elements shown in the process control model, namely goals, observed state, expected state, expectation discrepancy, inverse model, and forward model.

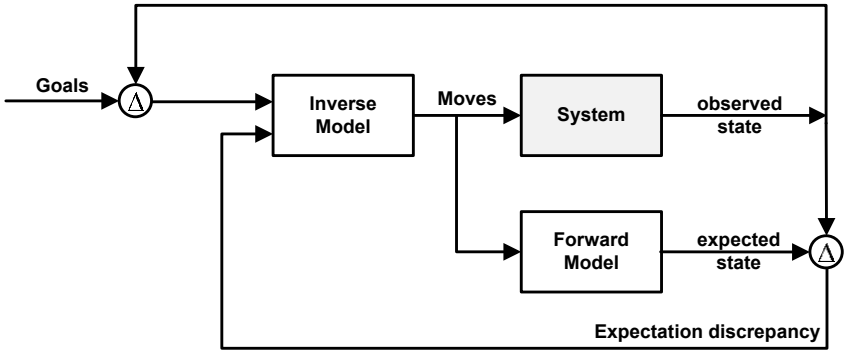

Figure 26. Process control model

The final output of a decision strategy is the action (move) that is applied to the system being controlled. In chronic disease care, the output is composed of treatment and scheduling moves (actions). All other elements are either input or intermediate outputs. A measurement of a patient state variable, such as blood glucose, is an example of an input from the intermediate move of information gathering. A feedforward strategy uses its forward model to generate an expectation for an action, and if this prediction is favorable, select that action as final output. In this case, the expectation is an intermediate output. The representation of a decision strategy must be sufficiently powerful to allow for this complex interplay between all these elements.

Figure 27 rearranges and simplifies the process control model (from Figure 26) and regards it at a given point in time as a single decision instance. This snapshot version of the process control model illustrates the question that this section addresses, namely, what needs to be inside the box labeled “decision strategy”, in other words, how to do we

represent the transformation of these inputs, through intermediate outputs, to the final output of an action?

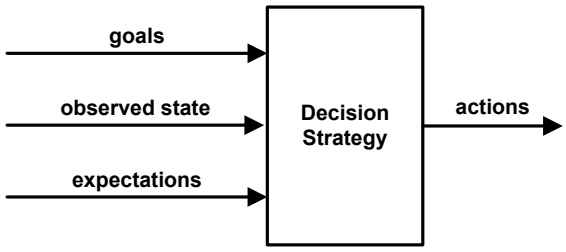

Figure 27. Input/output of decision strategy

The inputs, goals, expectations, and observed state for a decision strategy can be expressed in terms of a problem space. The observed patient state is a point in the state space. A goal describes a region of the state space, for example, all points where blood pressure is below 130mmHg. Expectations are generated by the mental model contained in the decision strategy as a prediction of where in the state space a patient will end up due to a given action. Below, it is shown how the decision strategy can be represented as a set of condition action rules containing functions that represent the mental model (expectations), observations, and externally provided goals, such as evidence-based goals.

The representation of a decision strategy proposed here is a set of *condition-action rules*, i.e., for a given patient state (a condition), a certain action will be taken. For example, "if a patient has LDL greater than 100mmHg, make a lipids treatment move". A condition-action rule can be represented as a tree; for example, the above rule is shown in

Figure 28. A rectangle represents a condition, a hexagon represents an action. Conditions and actions will be defined more precisely in the following subsection.

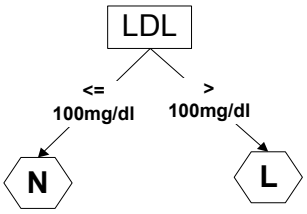

Figure 28. A simple strategy to treat lipids when LDL is above goal (L = make a lipid treatment move, N = take no action)

As defined in Section 1.2, a decision strategy must return an output (e.g., an action) for any input (e.g., patient state). Therefore, for a tree to represent a decision strategy, it must satisfy the following constraints: (1) Every decision path in the tree must terminate in an action, even if the action is "do nothing" (represented by **N** in Figure 28). In other words, every leaf of the tree must be a hexagon. (2) Arrows can only point from conditions to either actions or other conditions. It is not permitted for arrows to originate from actions. (3) Every decision strategy must contain at least one action.

The strategy shown in Figure 29 contains conditions that connect (with arrows) to other conditions, and as a result, the strategy prioritizes treatment so that it treats lipids to goal before treating glucose, and treats glucose to goal before treating blood pressure. (This is the "serial LGB" strategy from Study 1.) All strategies discussed in Studies 1 and 2 can be represented in this way.

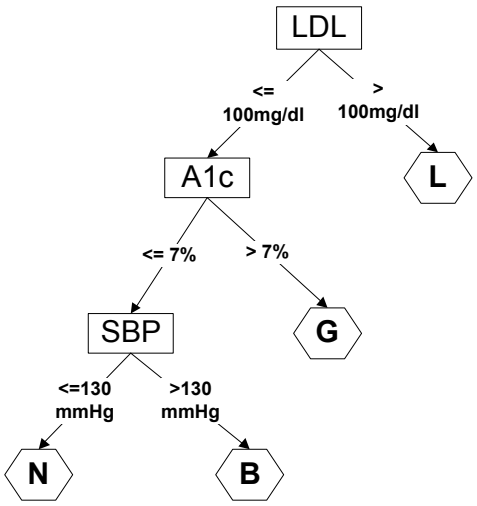

Figure 29. Tree representation of serial LGB

A detailed definition of this structure can be given as a context-free grammar in Extended Backus-Naur form in Appendix C (International Organization for Standardization 1996). We have now defined a structure for decision strategies but have not yet defined the conditions and actions. The nature of these building blocks is the topic of the following subsections.

Conditions

A *condition* is defined as a Boolean function of the patient state, i.e., it evaluates the patient state at a given point in time and returns either True or False. For example, a simple condition is "Is LDL greater than 100mg/dl?" An example of a more complex condition is "is the distance-to-goal for A1c greater than the distance-to-goal for SBP?" The first condition is represented in Figure 30 in two equivalent ways.

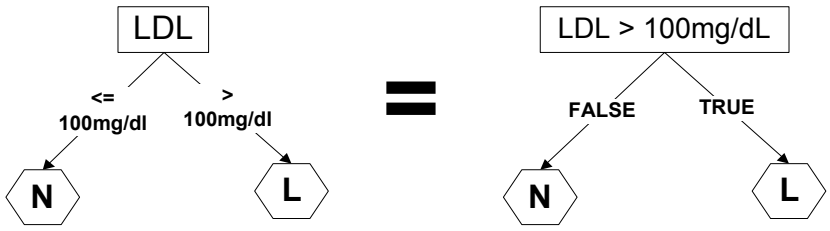

Figure 30. Two equivalent representations of the condition "LDL > 100mg/dl"

The power of a strategy to treat effectively is constrained by what state variables it has access to and how they are represented. For example, if a strategy does not have access to blood pressure information, it cannot assess whether treatment for blood pressure is necessary and take appropriate treatment action.

The representation of state variables must be useful for decision making. A straight comparison "is A1c greater than SBP?" does not make sense because of the differences in units of measure and ranges of the variables. However, representing these variables in terms of distance-to-goal or based on a distribution (as a percentile) makes it possible to compare such variables in meaningful ways.

Goals. Goals are presented as a goal function Goal(X) that returns the goal value for a given state variable. For example, Goal(SBP) = 130mmHg. Goals can be imposed externally or the goal function can be evolved.

Expectations / forward model. The forward model makes a prediction of a future state based on the current state and an anticipated action. If we consider a condition-action rule R that maps a state S to an action A, $R(S) \rightarrow A$, then, using the same notation,

a mental model can be stated as a function $Exp(S, A)$ that generates a new, expected state S'. Since the output of the mental model is again a state, a mental model can be represented as function in a condition-action rule because the expression $R(Exp(S, A))$ is a valid functional expression and can be evaluated. Therefore, the mental model can be represented by the function Exp(X, a), taking a state variable and an action as parameters. For example, if the current blood pressure is 140mmHg and the expectation of the next blood pressure move is a 5mmhg drop, then Exp(SBP, B) (read "the expected SBP given that a blood pressure move is made") will be 135mmHg. An expectation discrepancy is represented as the difference between an expectation and the current value of a variable.

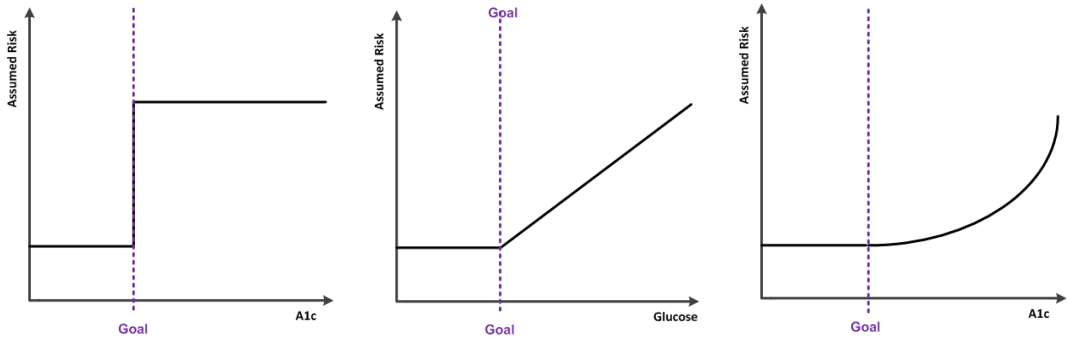

Figure 31. Three different expectations of the relationship between risk and A1c

Figure 31 illustrates three different functional forms of an Exp(X,a) function. It depicts the representation (mental model) of the relationship between risk and A1c. On the left, the relationship is a step function and a strategy using this expectation would focus on bringing A1c below goal in order to obtain a risk benefit. The middle graph shows a linear relationship and results in a strategy that assumes a risk-reducing benefit

for each glucose treatment. Finally, on the right, is an exponential expectation. A strategy using this expectation would give a higher priority to bringing A1c from 11 to 10 than from 8 to 7 because there is a greater payoff in terms of risk reduction (fitness).

The Exp(X,a) plays a critical role in determining whether a decision strategy is feedback or feedforward. The presence of the Exp(X,a) function in the conditions for a decision strategy is necessary for the strategy to be feedforward, i.e., it must generate and use predictions to guide its decision making. Conversely, absence of the Exp(X,a) function is sufficient for a strategy to be feedback.

A strategy updates its representation of the current value of a patient state variable by using *information-seeking moves* (e.g., blood pressure measurement, lab tests). This is represented by the Cur(X) function, read as "current value of X". Finally, the strategy may refer to a previous encounter (through some form of memory, such as a patient chart). This is represented by Prev(X), the previously recorded value of X, and PrevAct(), the action taken at the last visit.

The list of these functions is summarized in Table 17.

Table 17. Conceptual description of functions for conditions

Function	Description
Cur(X)	Current value of a state variable (age, A1c, SBP, creatinine, etc.)
Exp(X, a)	Expected value based on an action a
Goal(X)	Goal value for a state variable
Prev(X)	Value of a state variable at time t-1
PrevAct()	Last action (taken at time t-1)

The inverse model in Figure 26 is represented by a tree, as shown, for example, in Figure 29. This tree combines relevant information to generate moves by incorporating this information into the conditions of the tree. Table 18 illustrates relevant examples of the use of these functions in conditions.

Table 18. Examples of conditions

Condition	Functional Expression
Is the patient currently above goal for A1c?	Cur(A1c) > Goal(A1c)
Was the patient above goal at the last encounter?	Prev(A1c) > Goal(A1c)
Is the patient expected to be above the SBP goal if I make a blood pressure move?	Exp(SBP, B) > Goal(SBP)
Is the patient expected to have an A1c above 9 if I make a blood pressure move?	Exp(A1c, B) > 9
Is the current A1c below where I would expect based on the last move?	Exp(PrevA1c, PrevAct()) > Cur(A1c)

These functions are also subject to constraints to ensure that every X is filled with a state variable and a is always filled with an action. The evolution of these functions into more complex composite expressions are described in Appendix C.

We have now answered the question of what is in the "decision strategy" box in Figure 27. A decision strategy is represented as a tree of condition-action rules that expresses the contents of the inverse model, which includes patient state information, plus goals, expectations (generated by the mental model), and observations used in order to arrive at a given action.

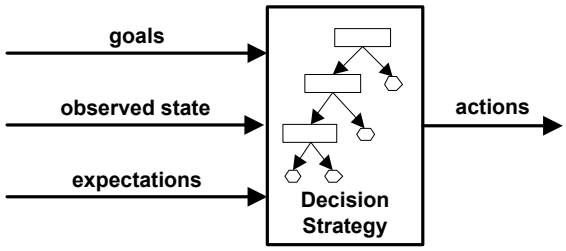

Figure 32. A decision strategy represented as a condition-action rule tree

Below, the nature of the actions is discussed in detail.

Actions

A treatment strategy makes three types of moves: (a) information gathering, (b) treatment, and (c) scheduling. Information gathering provides inputs, as discussed above. For the purposes of this study, it is assumed that information gathering moves, such as lab tests and blood pressure measurements, are made at every encounter to update the mental model with the current patient state. What remains to be defined is the output of a strategy – its decisions about what to treat and when to schedule the next visit.

In the experiments below, a strategy is considered to have five options for what to treat: (N) do nothing, (A) adherence, (G) glucose, (B) blood pressure, and (L) lipids. Once a condition is chosen, the specific treatment action (e.g., medication and dose to treat) can be selected, for example, from the clinical guidelines described in *Staged Diabetes Management* for that particular condition (Mazze et al. 2007; Mazze et al. 2012). *Staged Diabetes Management* is particularly practical for this purpose because it

provides a treatment algorithm for each condition that describes what medications to use, in which order, and in what steps to increment them sequentially to intensify treatment.

Instead of choosing particular treatment actions from a guideline, one can use the genetic programming approach to construct decision strategies that initiate particular treatments, for example, by letting it prescribe a specific dose of a particular oral agent. However, the trade-off is that this greatly increases the search space and the regions of infeasible actions (e.g., prescribing 804mg of Metformin, a dose that is not available in pill form). Consequently, a greater population of initial strategies or more generations are needed in order to find strategies with greater fitness, and the risk is greater that the resulting strategy is either hard to interpret or not feasible for practice.

For scheduling, the strategy selects the number of days from the current time until the next visit is scheduled. The action resulting from a strategy can be described as a tuple (T, d) where T is the type of treatment and d the amount of days until the encounter.

The choice of representation constrains the space of decision strategies that can be created. For example, the representation describes here allows only for random constants, not a function returning a random value. As a result, it is impossible to evolve a random choice strategy. However, as discussed in detail in Appendix C, the representation presented here is flexible and does not constrain the evolution of strategies in ways that allow only existing strategies to be evolved.

A decision strategy can now be represented as a tree of condition-action rules that uses patient state as input and generates actions (treatment and scheduling) as output. Genetic operators, such as cross-over and mutation, can be applied to manipulate this tree representation. This is discussed in the following section.

6.4. An Evolutionary Process to Manipulate Decision Strategies for Treatment of Patients with Type 2 Diabetes

Study 2 investigated the effect of selective attenuation or amplification on the cost-effectiveness - fitness - of existing strategies. In the present study, the effect of systematic manipulation of the strategy itself on its fitness is investigated. A strategy is changed by manipulating its representation as a decision tree. For example, Figure 33 demonstrates how a simple change, swapping the shaded node with the top node, changes a strategy from serial LGB to serial GLB. The tree representation can be considered the genome of a strategy; it encodes how the strategy uses patient state information to choose actions, including treatment and scheduling. This analogy allows us to bring to bear powerful tools for systematic manipulation, namely, genetic operators, such as crossover and mutation.

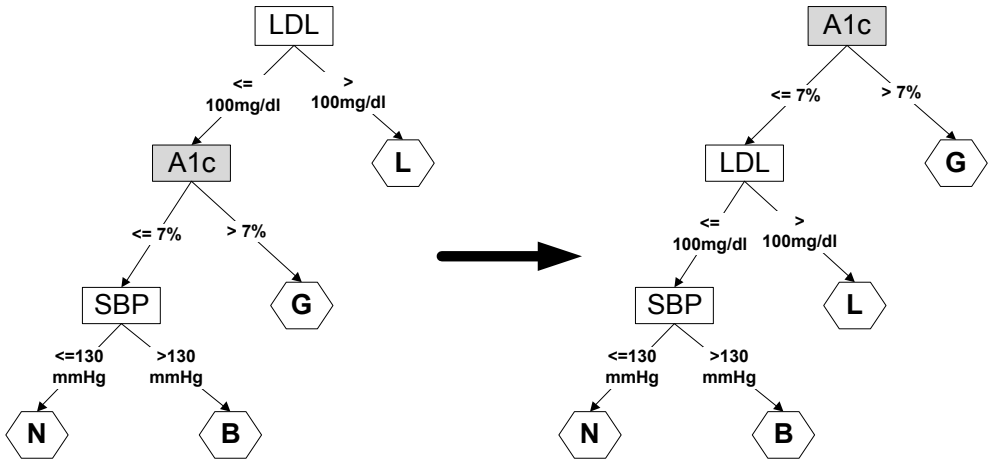

Figure 33. Changing a decision strategy by manipulating its representation: a single swap changes the strategy from Serial LGB to Serial GLB.

In evolutionary processes, individuals in a new generation are the product of either asexual or sexual reproduction. Asexual reproduction, from here on referred to as *reproduction*, creates an exact copy of the "parent". Sexual reproduction, from here on referred to as *cross-over* or *recombination*, creates a combination of the genetic representation from two "parents". In this fashion, the terms reproduction and cross-over used here are consistent with the use proposed by Koza (1992).

After a new generation is populated by reproduction and cross-over, every strategy in the new generation has a probability of being mutated, i.e., its genetic representation is slightly disturbed.

6.4.1. Initial Seeding (Generating the First Generation)

An initial set of decision strategies is generated by randomly assembling conditions and actions and combining them in a tree that satisfies the outlined constraints.

Conditions of the form $(X > Y)$ are generated by randomly selecting state variables, random constants, or functions for both X and Y. Actions are generated by randomly choosing a condition to treat and a scheduling interval. A condition or action is chosen as the root node for a tree. If the node is a condition, a condition or an action is chosen for the branch representing that the condition is true (i.e., X is greater than Y), and another for the "false" branch (X is less than or equal to Y). This process is repeated recursively for all nodes which are conditions. To keep the evolved strategies parsimonious as the decision trees grow larger, the choice whether an action or condition is selected for a given node is stochastically biased to favor actions, which limits the growth of the trees. The generation of a strategy ends once all leaf nodes are actions.

6.4.2. Reproduction (Asexual Reproduction)

From the set of strategies in a given generation n, each strategy is selected with a probability proportional to the strategy's fitness to be copied exactly into generation n+1. Consistent with selection, the strategies with the highest fitness – e.g., that achieve the greatest risk reduction – have the greatest likelihood of being preserved.

6.4.3. Cross-Over (Sexual Reproduction)

From the set of strategies in generation n, two are selected with probabilities proportional to their fitness. The selection process is discussed in more detail in Section 6.4.5 below. The selected strategies are the parents of a new strategy that will be created by randomly swapping tree nodes among the two strategies.

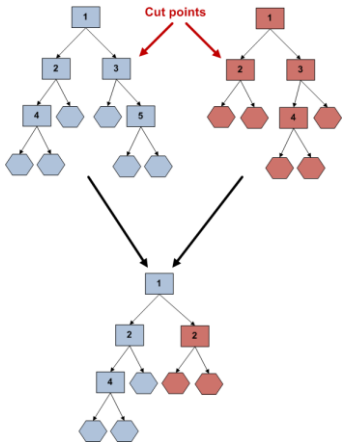

Figure 34. Example of cross-over

6.4.4. Mutation

The following types of mutation can occur: rule addition, rule deletion, sequence change, condition change.

Rule addition. A random condition-action rule is inserted into the tree, as shown in Figure 35.

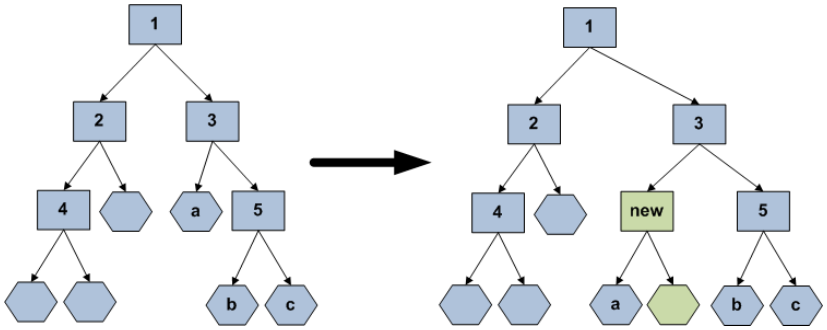

Figure 35. Example of rule addition mutation

Rule deletion. A random node is removed from the tree, as shown in Figure 36.

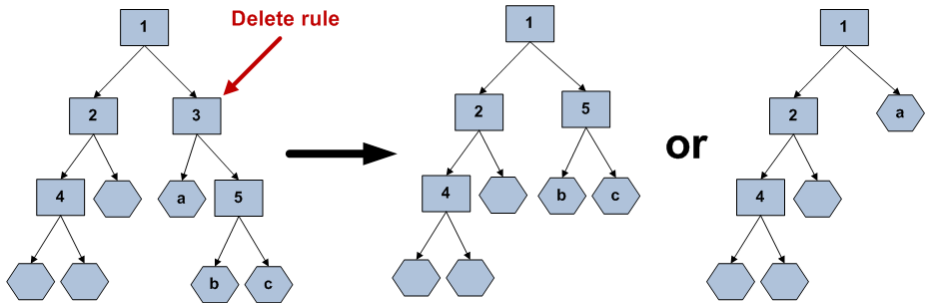

Figure 36. Example of rule deletion mutation

Sequence change. Two nodes of the tree are randomly swapped.

Condition change: operator swap. Conditions are expressed in the form $X > Y$. This mutation changes the places of X and Y, which has the same effect as swapping the operator.

Condition change: expression change. The expression of the left- or right-hand side of the condition is randomly changed.

6.4.5. Selection

Selection pressure is the degree to which strategies with greater fitness are favored for reproduction as opposed to allowing all strategies to reproduce (Miller & Goldberg 1995). Selection is accomplished by creating a *mating pool* which contains only a subset of the population of strategies for a given generation. In the research presented here, individuals are selected into the mating pool using *tournament selection* (Miller & Goldberg 1995). For tournament selection, k strategies are randomly selected from all the strategies in the current generation, and the best (greatest fitness) strategy for a given

tournament is added to the mating pool. The selection pressure varies with the size of tournaments, larger tournaments (greater values of k) result in higher selection pressure and therefore lower the chances of weaker (low fitness) strategies to survive. A tournament size of 1 would result in random selection of strategies regardless of fitness. A tournament size of k=5 was chosen to select against approximately 50% of the individual strategies in a generation while guarding against over-fitting and results being strongly influenced by stochastic noise (Blickle & Thiele 1996).

Once the selection for the mating pool has been completed, two strategies are randomly selected from the mating pool as parents for a new strategy. This is repeated until the next generation is populated with N strategy individuals. Complete extinction of a type of decision strategy occurs when no individual that contains the structure of that strategy is selected into the mating pool. For example, consider a strategy that contains a distance-to-goal rule, and through the evolutionary process, there are five individual strategies that contain this rule in a given generation. If none of these five are selected into the mating pool through selection, this rule will not be passed on to the next generation and becomes extinct.

6.4.6. Process

Given these genetic operators, the following process can be applied to evolve decision strategies:

(a) generate an initial set of strategies composed of random condition-action rule trees;

(b) assess the *fitness* of the strategies (a function of the path each strategy produces);
(c) select strategies to reproduce for the next generation;
(d) use a combination of the following methods for reproduction: copy selected strategies into the next generation as-is, combine two strategies (sexual reproduction) to form off-spring strategies, and mutate strategies in the new generation (with a small probability, introduce a random change);
(e) return to step (b) for the new generation of strategies.

To complete this process, the only piece left to define is fitness functions.

6.5. Fitness Functions

Fitness functions evaluate decision strategies with respect to how they move patients through the fitness landscape, i.e., their effect on risk reduction or cost-effectiveness. Greater fitness is the result of greater adaptation of the strategy to its environment.

The fitness function plays a role in the evolved strategies *adaptability*, its ability to succeed in a changing environment. This is closely related to the concepts of robustness and over-fitting. A strategy may evolve to perform exceptionally well in its current environment, but a minor change in the environment may cause it to fail. An adaptable strategy, on the other hand, is more robust because it is not sensitive to small changes in the environment.

Below, the different types of fitness functions are discussed.

6.5.1. Risk-Based Fitness

The use of risk as a fitness function is particularly elegant since risk is closely linked with survival. Survival is a key concept in genetic programming because it determines which evolved structures – in this case, decision strategies – are carried forward to the next generation and which ones become extinct due to lack of fitness. Moreover, traditional risk models for all adverse clinical events, fatal and non-fatal, use survival analysis and investigate the question of how long a patient will survive without experiencing a given event (Bland 1995). In this case, the fitness value for a given patient is determined directly from the risk estimate using that patient's state. However, strategies are unable to address the irreversible portion of risk (which is, for example, due to age and gender) and should not be selected against because they are unable to affect this risk. Therefore, we formulate a fitness function based on reversible risk, which considers fitness as the difference between the risk for the current patient state and the lowest risk that can be attained with treatment. The total 10-year risk of adverse events for a given patient is calculated using the UKPDS Risk Engine (Stevens et al. 2001). Using the Simcare model, treatment of all conditions for this patient is simulated and the resulting patient state is used as input for the UKPDS Risk Engine to estimate the irreversible risk, i.e., the risk due to factors such as age, gender, and ethnicity. This number is subtracted from the total risk to obtain the given patient's reversible risk.

The metric used in this study to assess reduction of reducible risk is the *number of prevented events*. For example, a baseline treatment strategy B (such as "no additional treatment), may result in a number of events E_B during the follow up period after

treatment, while another strategy S results in E_S events. The number of events prevented by S (as compared to B) is $E_B - E_S$. As in Study 1, we choose "no additional treatment" as a baseline, which implies that no other strategy should have fewer events unless it is a strategy that actively causes harm.

An example of a reversible risk fitness function f is shown below. Let X_i be a patient state vector for patient i, and $G(X_i)$ the hypothetical patient state for patient i if i were treated to goal for all conditions. Let $R(X_i)$ be a function that returns a risk estimate for a given patient state X_i. Let $\mathbb{X}$ represent the population of n patients as described as a set of patient states $\mathbb{X} = \{X_0, \dots X_{n-1}\}$. $f(S, \mathbb{X})$ is a function that returns fitness of a strategy S for the population $\mathbb{X}$.

$$f(S, \mathbb{X}) = \frac{1}{n} \sum_{i=0}^{n-1} \Big(R(X_i) - R\big(G(X_i)\big) \Big)$$

6.5.2. Cost-Based Fitness

Cost-based fitness is based on the idea that decision strategies are penalized for the consumption of resources. This is consistent with the evolutionary advantage of life forms that use fewer resources than their competitors – they have a greater chance of survival, especially in environments where resources are scarce. In the diabetes care context, cost would be a contributor to overall fitness rather than make up the entire function; otherwise treatment strategies could be selected for just by being very inexpensive, however without producing satisfactory outcomes (e.g., "no additional treatment").

For the purpose of this Study, cost-effectiveness is defined in terms of the treatment cost needed to prevent an additional event, i.e., $\frac{C_S - C_B}{E_B - E_S}$, the additional cost of treatment for a strategy S compared to a baseline strategy B divided by the number of events prevented by strategy S compared to B (Gold et al. 1996). As above, the strategy "no additional treatment", defined the same as in Study 1, is used as baseline B. The denominator is the number of events prevented, defined as above in the discussion of risk-based fitness.

For example, if the baseline strategy results in 320 events and strategy S results in 290, but S incurs an additional $50,000 in treatment costs (compared to the baseline), the cost-effectiveness ratio for S is $50,000/(320-290) = $1,666.67 per prevented event. In other words, additional annual treatment expenses of $1,666.67 are needed to prevent one additional event in ten years. This strategy would be equally cost-effective as another strategy S' which prevents 60 events for $100,000 additional treatment costs because they have the same cost-effectiveness ratio.

There are two special cases to consider, namely, cost-saving strategies and harmful strategies. A strategy is *cost-saving* if $C_S < C_B$ and $E_S < E_B$, i.e., it results in lower costs and prevents more events than the baseline strategy. A cost-saving strategy results in a negative cost-effectiveness ratio. A strategy S is *harmful* if $E_B < E_S$, i.e., it results in more events than the baseline strategy. Such a strategy is assigned ∞ as cost-effectiveness ratio to ensure harmful strategies are selected against. Otherwise, the harmful strategy could appear to be cost-saving, for example, a strategy that incurs

$50,000 in treatment costs but results in 30 additional events would have a cost-effectiveness ratio of ($50,000)/(320-350) = -$1,666.67.

6.5.3. Complexity-Based Fitness

In addition to outcomes, fitness can be based on characteristics of the evolved strategy. Especially for decision strategies, parsimony of rules may be advantageous for ease of interpretation and application. Furthermore, limiting complexity guards against over-fitting a strategy to the specific problem space in which it has evolved. A strategy may have adapted perfectly to the current environment but lack robustness against changes (Forster 2000), in other words, it lacks adaptability.

Imposing restrictions on the number of rules is similar to the pruning of decision trees (Quinlan 1987). In general, in genetic programming, adaptive pressure towards parsimony has been found to result in more elegant solutions with lower complexity than what evolved without that pressure (Banzhaf et al. 1998). This parsimony requirement can be added as a fitness criterion by incorporating a penalty for strategy length (number of condition-action rules) into the fitness function.

For example, the reversible risk fitness function from Section 6.5.1 can be modified by adding a weight function w that that assigns values between 0 and 1 based on cardinality of the number of rules contained in strategy S. For example, a fitness penalty can be assigned for every additional rule by adding a weighting function $w(x)$ that returns decreasing weights for larger x. An example of a fitness function with complexity penalty is shown below.

$$f(S,\mathbb{X}) = \frac{w(|S|)}{n}\sum_{i=0}^{n-1}\Big(R(X_i) - R\big(G(X_i)\big)\Big)$$

6.6. Experiments

Using the process discussed above, decision strategies can be evolved using a genetic programming approach. This method is used to generate decision strategies for the prioritization of care problem and evaluate their performance in terms of fitness (based as risk or cost-effectiveness, depending on the experiment). Suppose a strategy evolves that achieves better (or worse) performance for a particular subgroup of patients than strategies examined in Studies 1 and 2. Examination of the evolved decision rules provides insight into how this strategy selects moves that result in success for this subgroup of patients.

It is expected that the genetic programming approach will evolve some strategies that are substantially like those described in Studies 1 and 2, for example, a version of a serial treatment strategy described in Table 7. This provides external validity to the evolutionary approach. If all the evolved strategies are unlike strategies observed in practice, this warrants further investigation because it could imply that the environment or fitness function are inappropriate for the problem or the necessary building blocks for existing decision strategies are not available.

6.6.1. Methodology

The genetic programming approach used in the presented research was implemented in C#. The system represents decision strategies as executable C# computer programs

that can be evolved using genetic programming. The resulting programs call the Simcare patient simulation software to gather information (measure patient state), apply treatment actions, and schedule future visits (McCabe 2012).

Following the approach outlined by Koza, a number of set sizes between 10 and 1,000 initial strategies were used and it was determined that there was no significant benefit from using sets of initial strategies greater than 100 in the three experiments (Koza 1992). Therefore, for each experiment, 100 initial strategies were randomly generated as described in Section 6.4.1. After every generation, fitness was assessed and strategies for the next generation were chosen by *tournament selection* (Miller & Goldberg 1995). The detailed configuration of the genetic programming approach, including the minor parameters described by Koza, is discussed in Appendix C.

The genetic programming approach is used in three experiments. The first two are designed to demonstrate that this approach generates solutions in spaces explored in Studies 1 and 2. The third experiment provides a more complex problem environment by adding the factor of adherence. Table 19 characterizes the key differences in environments for the three studies and illustrates that each successive study incorporates the problem space of the previous studies.

Table 19. Problem environments for Studies 1, 2, and 3

	Scheduling 14-89 days	**Scheduling 90 days and more**
High adherence	Studies 2 and 3	Studies 1, 2, 3
Low adherence	Study 3	Study 3

6.6.2. Experiment 1: Reducing Risk in Prioritization of Care

Experiment 1 uses the genetic programming approach in the same context as Study 1, namely, prioritization of care which allows a single opportunity to treat at every encounter, having all encounters spaced 90 days apart. The objective in this context is to reduce risk; therefore, a risk-based fitness function is used.

Table 20. Tableau for Experiment 1

Parameter	Value
Objective	Find a strategy that maximizes risk reduction for patients
Terminal set	Actions: (B, 90), (G, 90), (L, 90), (N, 90)
Function set	As described above (including tree representation and Table 17)
Fitness cases	18,356 simulated patients (with characteristics described in Section 3.3)
Raw fitness	Total event rate after one year of treatment
Standardized fitness	Total rate of events prevented for population after one year of treatment
Initial number of strategies	100
Generations	101 (incl. initial generation 0)

6.6.3. Experiment 2: Increasing Cost-Effectiveness of Treatment

Experiment 2 mirrors Study 2 with its focus on cost-effectiveness. As in Study 2, strategies are given additional degrees of freedom by varying scheduling. The selection of strategies uses a cost-effectiveness fitness function in this experiment.

Table 21. Tableau for Experiment 2

Parameter	Value
Objective	Find a strategy that maximizes cost-effectiveness
Terminal set	Actions: (B, d), (G, d), (L, d), (N, d) where d may be any number of days between 15 and 180.
Function set	As described above (including tree representation and Table 17)
Fitness cases	18,356 simulated patients (with characteristics described in Section 3.3)
Raw fitness	Cost-effectiveness ratio (Chapman et al. 2000)
Standardized fitness	Cost per prevented event
Initial number of strategies	100
Generations	101 (incl. initial generation 0)

6.6.4. Experiment 3: Treatment under Conditions of Varying Adherence

In Studies 1 and 2, and Experiment 1 and 2 of this study, patient adherence was assumed to be high and constant. Patient adherence refers to how well a patient complies with prescribed treatments. For example, if a patient has a single oral medication prescribed, and takes pills on 27 out of 30 days in a month, that patient is 90% adherent. A patient may also fail to adhere to dietary restrictions, as is common in the holiday season. Patients may not adhere to treatment for a number of reasons, including avoidance of side effects, forgetfulness, depression, and economic reasons. Some reasons for low adherence (for example, seasonal adherence) have discernible patterns that can be detected, however, others, such as forgetfulness, appear random.

Adherence cannot be measured directly. For some treatments, such as oral medications, adherence can be inferred from fill rates; if patients do not fill medications, they cannot take them. Lack of adherence can also sometimes be determined in discussion with the patient, or inferred from treatments not generating the expected response.

In Experiment 3, different assumptions with respect to adherence are examined to provide additional variation in the environment in which strategies evolve and determine its effects of risk. Three scenarios are considered: a population of high-adherence patients (comparable with Studies 1 and 2), a population of low-adherence patients, and a mixed population.

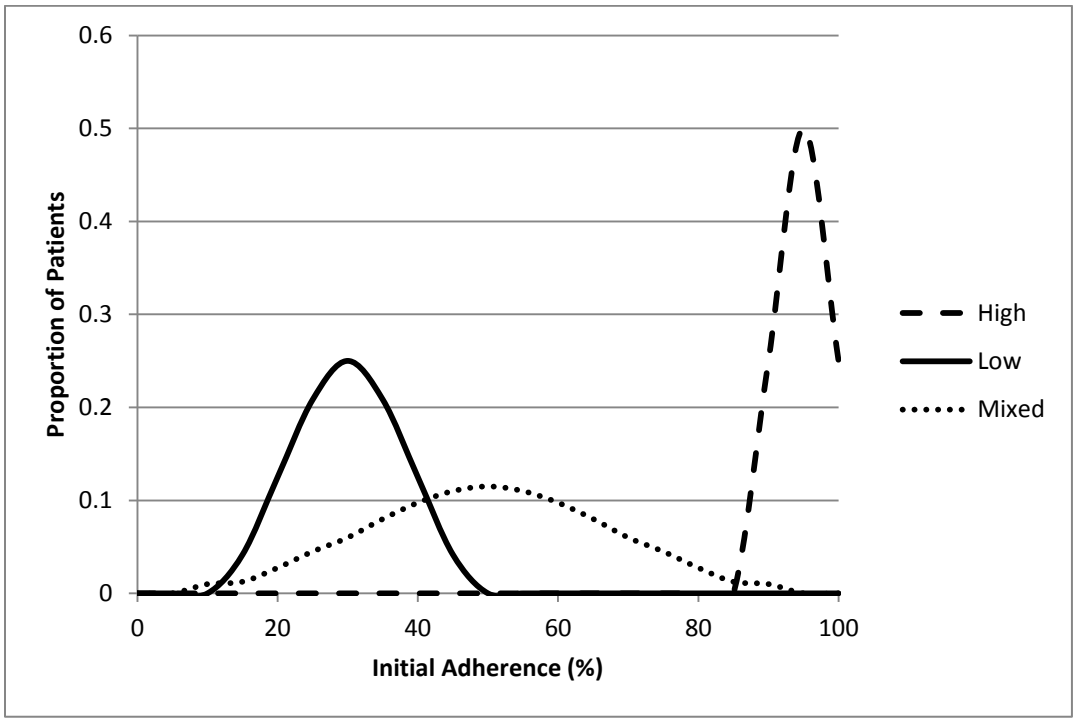

Figure 37. Initial distribution of adherence in the patient population for high (dashed), low (solid), and mixed (dotted) adherence environments

Table 22. Tableau for Experiment 3

Parameter	Value
Objective	Find a strategy that maximizes risk reduction for patients
Terminal set	Actions: (A, d), (B, d), (G, d), (L, d), (N, d) where d may be any number of days between 15 and 180.
Function set	As described above (including tree representation and Table 17)
Fitness cases	18,356 simulated patients (with characteristics described in Section 3.3)
Raw fitness	Total event rate after one year of treatment
Standardized fitness	Total rate of events prevented for population after one year of treatment
Initial number of strategies	100
Generations	101 (incl. initial generation 0)

6.7. Results

6.7.1. Experiment 1

In this experiment, strategies were evolved under the same constraints as in Study 1. We are specifically interested in the worst strategy evolved, which performed worse than random choice, the best strategy, and strategies comparable to those observed in practice.

Worst Strategy

In the first five generations, 16% of the randomly generated strategies performed worse than random choice. The worst such strategy is depicted in Figure 38. It treats only blood pressure, and only for patients below age 55, otherwise it takes no action. This is an example of a strategy that developed an error of omission, i.e., it fails to treat patients that are above evidence-based goals (Ramsey 2010). It performs poorly in terms of fitness because there are more opportunities to reduce risk in older patients, as shown in Figure 25. In contrast, the random choice strategy has no principled approach but will always choose an action and result in treatment for every patient.

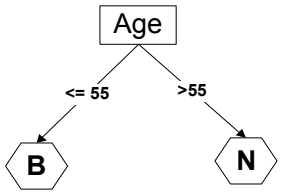

Figure 38. Worst strategy evolved during early generations of Experiment 1

However, the last strategy that performed worse than random choice was eliminated in the 10th generation.

Best Strategy

Within the constraints adopted for this research, the evolutionary approach converged on a stable best strategy S^* that first appears in its complete form in generation 17 and remains in the population for the remaining generations. S^* showed performance in reducing risk comparable to the post-hoc optimal choice strategy from Study 1. The key events in the pathway which led to the evolution of this strategy are illustrated in Figure 39 and summarized in Table 23.

Through reproduction, the genetic code of the strategy S^* spread throughout the population in the following generations. Mutations resulted in minor variations of S^*, for example, in generation 32, the second condition changed from "A1c > 8.1" to "A1c > 10.8". However, this mutated version of S^* was not as effective at reducing risk and, therefore, selected against in generation 36.

Table 23. 10-year cardiovascular event rate for benchmarks and evolved strategies

Strategy	Generation	10 year event rate after treatment (raw fitness)	# of events prevented (standardized fitness)
Benchmarks			
(-) No additional treatment		.320	0
(#) Random Choice		.294	26
(+) Optimal (post hoc)		.258	62
Evolved Strategies			
Initial Strategy (A)	0	.272	48
Initial Strategy (B)	0	.292	28
Initial Strategy (C)	0	.278	42
(A+B) Offspring of (A) (B)	2	.269	51
((A+B)+C) Offspring of (A+B) and (C)	11	.263	57
((A+B)+C)' Mutation of ((A+B)+C)	17	.259	61

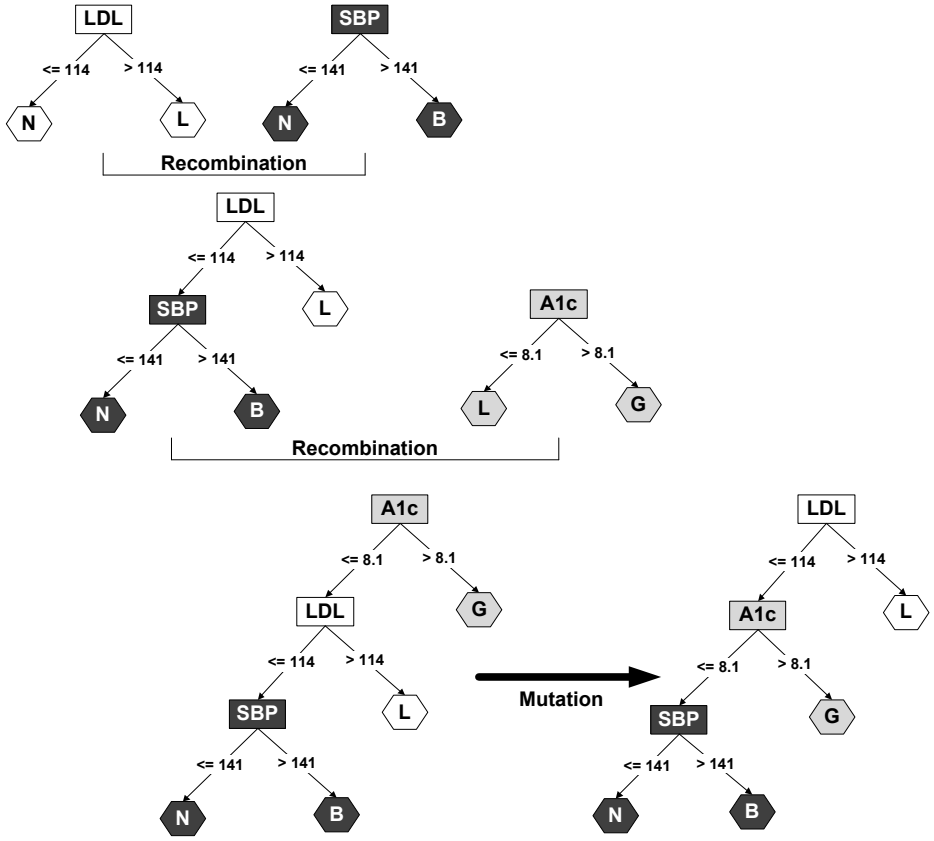

Figure 39. Evolutionary pathway to the best strategy in Experiment 1

The strategy resulting from this pathway is a serial treatment strategy that first focuses on lipids, then glucose, then blood pressure. It has slightly different goal thresholds than the "Serial LGB" strategy in Study 1, but an identical structure. As discussed above, it is expected (and contributes to external validity) that the evolutionary approach produces some strategies that have been developed by other means, similar to Koza's experiments producing patented circuit designs (Koza et al. 2003).

Strategies Matching Existing Practice

Another example of a strategy that corresponds to the observed strategies is shown in Figure 40. This strategy, which evolved in generation 9, is a distance-to-goal strategy that treats blood pressure if it has a greater distance-to-goal than lipids, otherwise will focus on glucose if it is above A1c of 8.8%, or lipids if below.

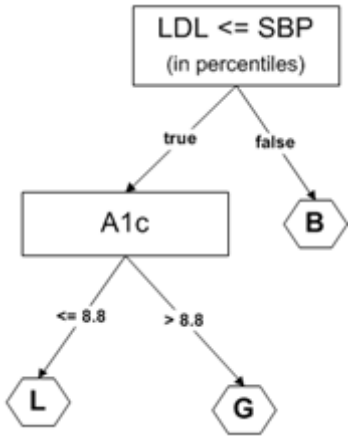

Figure 40. Examples of an evolved distance-to-goal strategy

6.7.2. Experiment 2: Effect of Cost-Effectiveness Fitness

Experiment 2 focused on cost-effectiveness and did not constrain scheduling to 90 day intervals but let strategies schedule visits in the range of 14 to 180 days. Fitness was defined by the treatment cost per prevented event, i.e., strategies that prevented more events with fewer treatment expenditures were selected for. The performance of the best strategy in terms of risk and cost is shown in Figure 41.

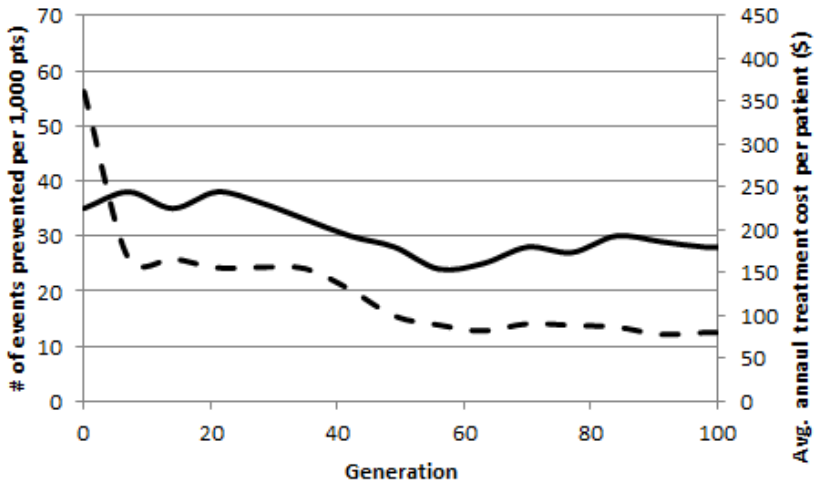

Figure 41. Prevented events (solid) and additional cost of treatment (dashed) in response to cost-effectiveness fitness function

The best strategy, as determined by the cost-effectiveness fitness function, is shown in Figure 42. One of its key features is that it partitioned patients by gender and age, and did not initiate treatments for male patients below age 55 or female patients below age 64. The first rules that partitioned patients by age emerged in generation 22. In generation 37, a gender rule was introduced through mutation. A subsequent cross-over brought this rule to the top of the tree. In generation 51, a cross-over resulted in essentially the same sub-tree being placed on both sides of the top node, which is still recognizable in the final strategy in Figure 42. Further mutations changed the LDL threshold later on for males (from 112 to 105), and introduced another rule to change treatment if the patient has a history of atrial fibrillation, resulting in the tree shown below.

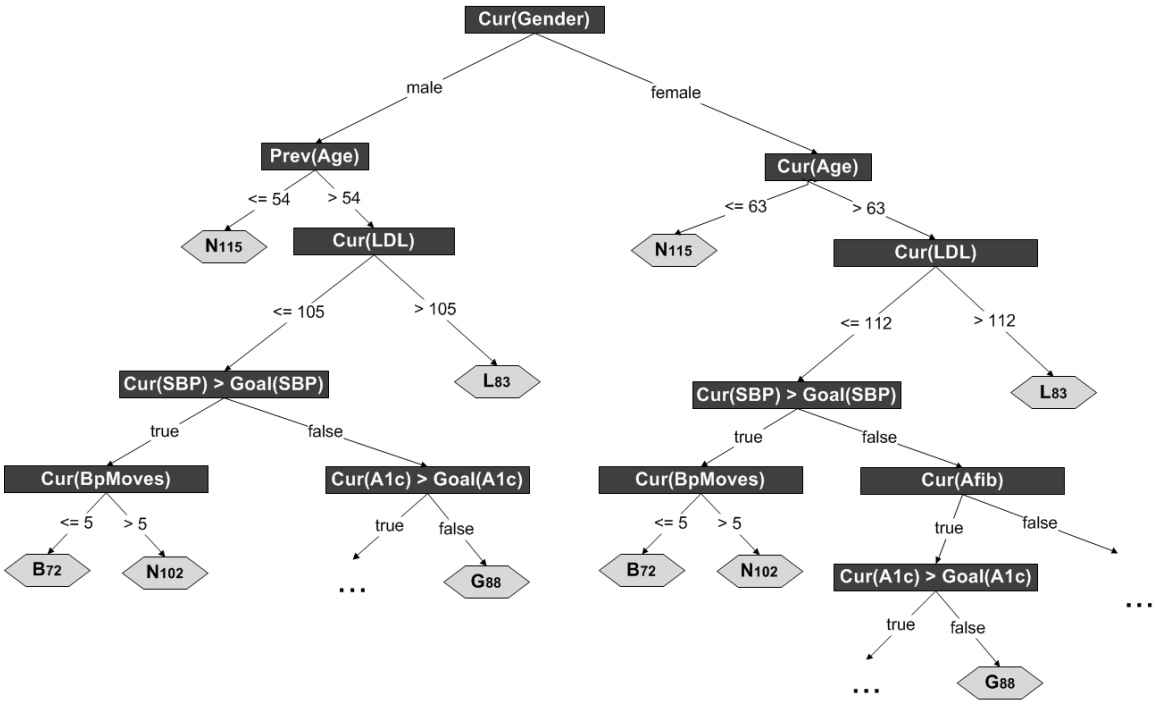

Figure 42. Best strategy in Experiment 2

The rules which partitioned patients by gender and age provided a fitness advantage and, through recombination, propagated throughout the population of strategies after generation 51. In the final generation, all surviving strategies contain rules that discriminate by age and/or gender. The fitness advantage with respect to cost-effectiveness is a result of the fact that there is a limited amount of reversible risk in younger patients and females. For example, a 45 year old male's heart attack risk can be lowered from 9.5% to 5.6% with statin treatment, but the same treatment will lower risk from 34.2% to 21.5% for a patient of age 70, all else being equal (Stevens et al. 2001).

The finding is consistent with similar cost-effectiveness research in the medical literature. For example, Denton et al. examined the optimal time to start statin therapy

using Markov decision process modeling and found that age and gender are important factors in this decision (Denton et al. 2009) and that it is not cost-effective to initiate statin treatment for some patients below a certain age. They determined cost-effectiveness by assigning a reward value (e.g., $50,000) to every quality-of-life adjusted year, deducting the cost of treatment from this reward, and calculating the lifetime reward for a patient. This approach requires an estimate of the patient's life expectancy that was not available for the study presented here. Instead, we calculated cost-effectiveness using the ratio described in Section 6.5.2, i.e., additional cost of treatment to prevent one additional event (compared to "no additional treatment").

The Denton study also used the UKPDS Risk Engine (as well as two other risk models that are not specifically designed for diabetic patients, Framingham and Archimedes.) They noted that the choice of risk model greatly influences the result, and can change the optimal start time to initiate statin treatment by up to 18 years. According to the UKPDS Risk Engine, *ceteris paribus*, risk increases exponentially with age, and the risk for a male patient is higher than that of a comparable female patient. The UKPDS Risk Engine is used to calculate the risk portion of the fitness function used in this study, as discussed in Section 6.5.1. The resulting fitness function implies that the cost of treatment outweighs the benefit in terms of risk reduction for some patients, especially those who are younger or female, i.e., the additional cost of treatment for these patients is not matched by a sufficient number of prevented events. The evolved strategies adapted to this fitness landscape and, in response, evolved behaviors that act as though they were

trying to achieve a minimum number needed to treat and, therefore, neglected subgroups of patients which would increase this number.

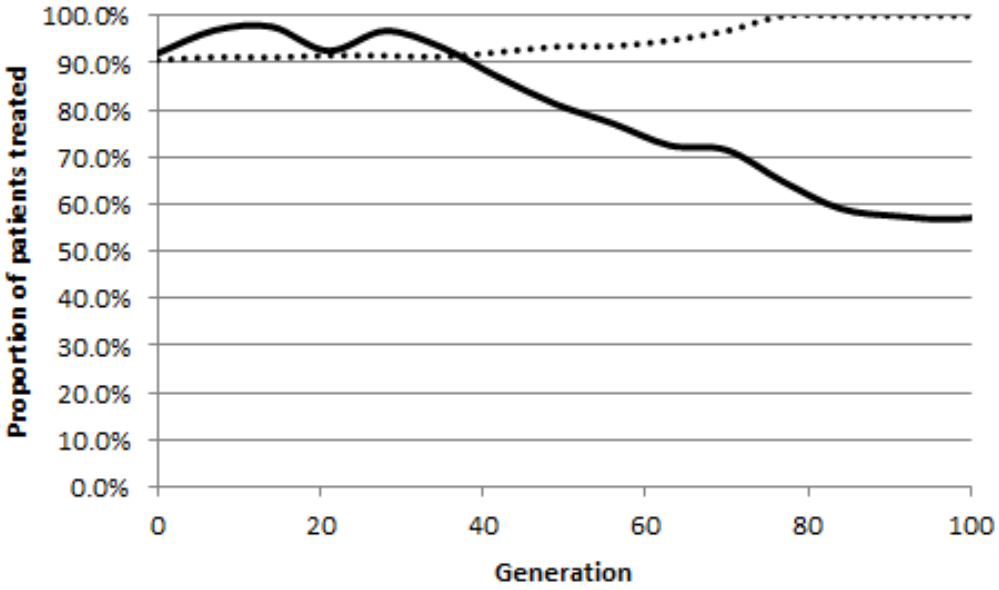

Figure 43. Proportion of patient treated when using a risk-based (dotted) vs. cost-effectiveness-based (solid) fitness function

To test that this behavior is indeed due to the cost-sensitive environment, another set of strategies was evolved using the same constraints and a (cost-insensitive) risk-based fitness function. When cost is not factor, "do nothing" actions are eliminated from the genetic material over time because they never contribute to risk reduction. Figure 43 shows how the proportion of patients who receive no treatment declines in the purely risk-based environment but increases in cost-sensitive environments. This supports that the discrimination behavior is indeed an unintended consequence of the cost-sensitive environment. It is noteworthy that this behavior is not a function of enlightened self-interest, as would be the case if it stemmed from rational economic agent that was trying

to optimize for a reward. Instead, it is the result of selection pressure in a cost-sensitive environment.

6.7.3. Experiment 3: Effect of Adherence on Evolved Strategies

The results for the best strategies after 100 generations in the different adherence environments are summarized in Table 24.

Table 24. 10-year cardiovascular event rate and mean number of moves during one-year treatment period for best strategies

Strategy	Moves per year	10 year event rate after treatment	Adherence moves per year	Glucose moves per year	Lipids moves per year	BP moves per year
Best Strategy: High adherence	6.4	.234	0.0	2.9	1.8	1.7
Best Strategy: Low adherence	7.7	.241	0.9	4.1	1.5	1.2
Best Strategy: Mixed adherence	9.3	.244	1.4	4.7	0.9	2.3

Strategies that evolved in the high-adherence environment do not consider adherence. By generation 15, all rules relating to the treatment of adherence had disappeared from the genetic material. In this environment, treating adherence is wasteful and does not provide a fitness advantage, so strategies that include adherence moves are selected against. As a result, the best strategy that evolved is essentially the same as a strategy that evolved in an environment where no adherence treatments are available (e.g., Experiment 2).

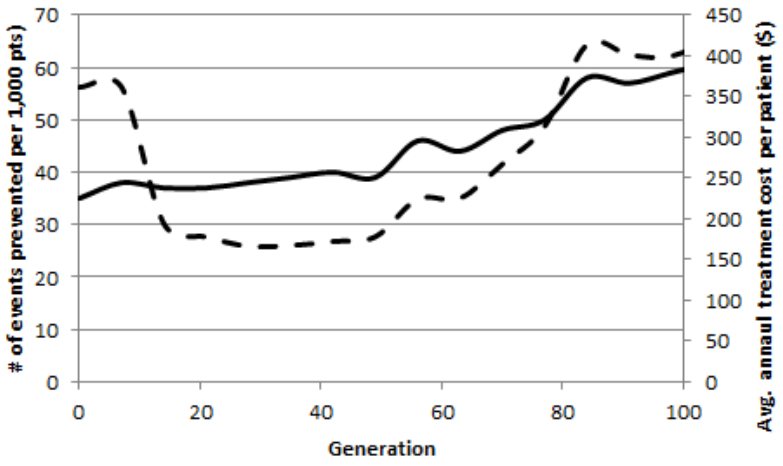

Figure 44. Prevented events (solid) and additional costs of treatment (dashed) for the best strategy in the high-adherence environment

Conversely, treating adherence in the low-adherence environment is always beneficial because all other treatments have very limited effects. Consequently, the most successful evolved strategy consistently assigned a high priority on the treatment of adherence. As seen in Figure 45, the number of patients treated for adherence rises rapidly in each generation in the low adherence environment and after 100 generations reaches 95% of the population.

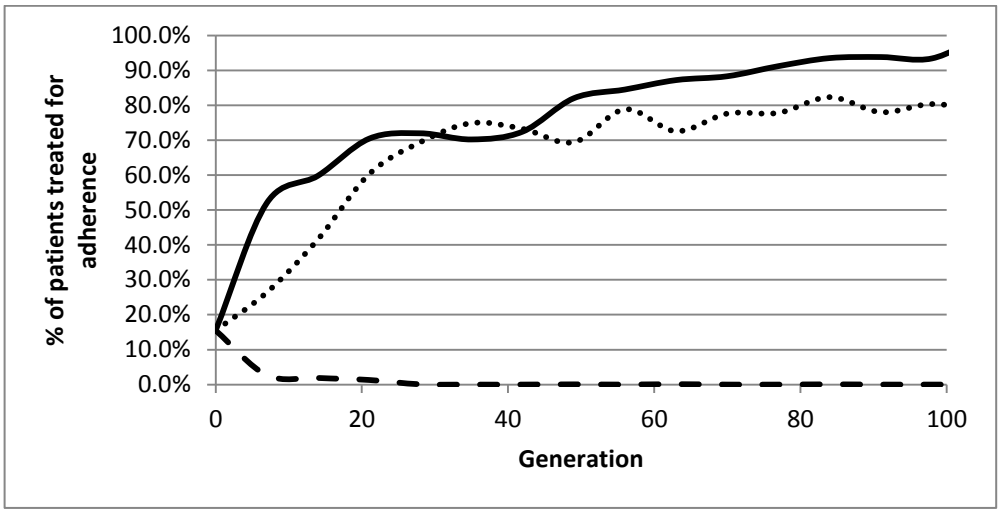

Figure 45. Proportion of patients treated for adherence in low (solid), high (dashed), and mixed (dotted) adherence environments

In the case of a mixed adherence environment, two types of strategies evolved successful behaviors, a heuristic feedback strategy and a feedforward strategy that considered expectation discrepancies as a lack of adherence indicator.

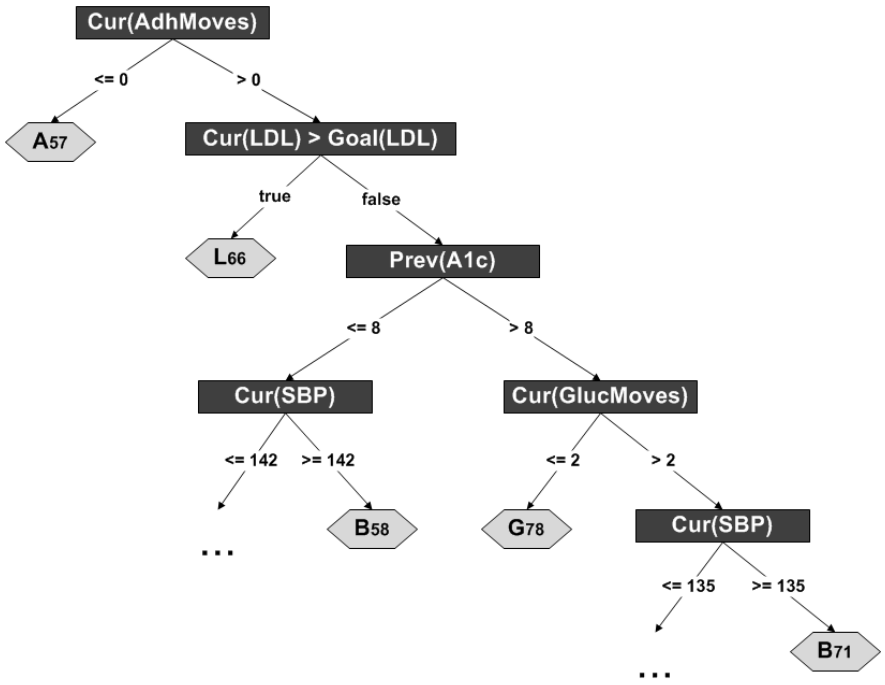

Figure 46. Feedback strategy for mixed-adherence environment

In generation 12, a heuristic feedback strategy evolved that prescribed adherence treatments to patients above LDL goal before initiating other treatments. This rule increased fitness and was incorporated into offspring, until ultimately through mutation and recombination, a rule emerged in generation 33 that treated all patients for adherence before initiating other treatments. This strategy is shown in Figure 46. The rule of interest is in the top node and checks if the strategy has already treated adherence, and if not, makes an adherence move. Consequently, the first move this strategy makes for a patient is an adherence move, regardless of whether the patient is already adherent or not.

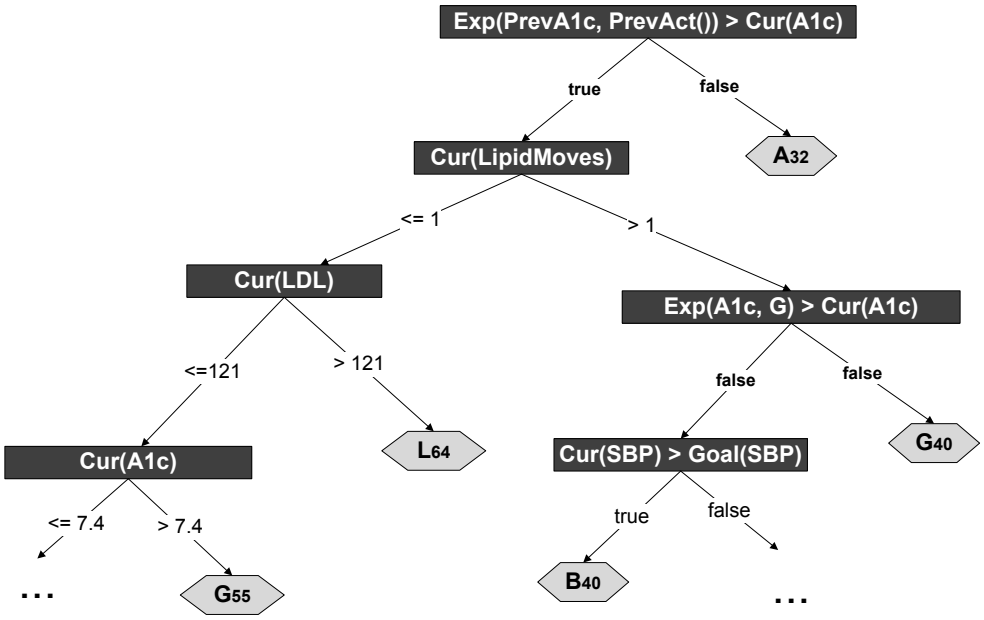

Figure 47. Feedforward strategy for mixed-adherence environment

The other successful strategy, which turned out to be the best strategy at the end of the run (shown in Figure 47), emerged in generation 91. It evolved a rule to discover an adherence problem using feedforward control with a simple mental model (containing the same expectations as in Study 1). This strategy checks if blood glucose was lowered as much as expected from the previous treatment action. If the previous treatment action was not a blood glucose move, there would be no expectation of a drop and no adherence action would be initiated. However, if a glucose move was made but did not achieve the expected drop, the patient would be treated for adherence. In other words, the strategy compares the expected improvement to the actual improvement, and treats adherence when there is a discrepancy. This strategy outperformed the feedback strategy which treated everyone for adherence, because it did not need to spend moves on adherence

treatment for patients that were adherent from the start. Furthermore, this strategy compensated for varying adherence while the feedback strategy treated adherence once and then treated as though it remains high permanently.

As shown in Table 24, the best strategy in the mixed-adherence environment required the greatest number of moves in a year. This is, in part, due to the fact that the strategy engaged in "experiments", i.e., it prescribed treatments and checked if the patient adhered and compensated, if necessary. Also, because the strategy was feedforward, it made more aggressive use of scheduling and placed greater emphasis on glucose and blood pressure moves than the strategy which evolved in the high adherence environment. The latter strategy used fewer moves and placed greater emphasis on lipids, which obtained a greater risk reduction than treating glucose or blood pressure with a limited amount of visits.

7. Discussion

Through a series of three studies, this research provides insight into the means that enable decision strategy performance. Specifically, it was found that performance is enabled by strong mental models that accurately represent a strategy's environment and allow prediction therein, or by the strategy being well-adapted and therefore able to exploit the structure of its environment.

Study 1 compares the performance of decision strategies in the context of the prioritization of care problem. These strategies differ in terms of the information they use for decision making; the computations they perform to select actions (inverse model); their expectations of effects of action (forward model); and feedback vs. feedforward control. One result from Study 1, demonstrating the performance benefits resulting from feedforward control with a strong mental model, is further explored in Study 2, while another result in Study 1, the discovery of ecologically rational strategies, is the focus of Study 3.

In Study 1, it was found that a feedforward strategy with a strong mental model that accurately predicts the consequences of its actions achieves performance that is comparable to (i.e., not significantly different from) an idealized optimal strategy. As expected from the dynamic decision making literature, this strategy is sensitive to the accuracy of the mental model. Even minor changes to the mental model can degrade the performance to being no better than random choice. This degradation of performance is to some extent the result of making errors; a strategy either fails to take the right actions

(i.e., makes errors of omission), or takes the wrong actions (i.e., makes errors of commission). Therefore, the performance of a strategy should be improved when these errors are identified and eliminated or reduced. This is the impetus for Study 2.

In Study 2, additional complexity is introduced to the prioritization of care context by allowing arbitrary instead of fixed scheduling and allowing more than a single move at a given visit. This additional flexibility allows for greater variation in performance between feedback and feedforward strategies and weak versus strong mental models. During this context, an approach named Procedo is developed to improve existing strategies by introducing "learning" from past failures and predicting them in the future. This approach creates more powerful strategies by adding anticipatory (feedforward) control to predict and compensate for errors. In every iteration, the proposed approach updates the strategy's mental model by adding a new prediction rule based on empirical data from past treatments. The results from Study 2 show that feedforward strategies resulting from Procedo succeed in complex cases where feedback strategies fail. This is consistent with observations from practice which show that physicians engaging in feedforward thinking are more successful in treating complex patient cases (Ramsey 2010).

The improved strategies in Study 2 develop mental models that contain more information about the relationship between risk and cost impacts of moves. Procedo-improved strategies become increasingly like the successful risk-minimizing feedforward strategies from Study 1 by identifying relevant features of the environment (patients) for

which a given move will be effective. In the case of Study 2, this results in the identification of relevant patient subgroups. For example, if the objective of treatment is to bring patients to goal within one year, a feedforward strategy can anticipate that six moves may be insufficient to bring patients with high blood pressure (i.e., patients in a subgroup with SBP greater than 160mmHg) to goal and schedule additional visits or initiate more than one blood pressure medication at a visit.

Study 2 develops a specific approach to modifying a strategy's mental model in order to improve the strategy's performance. This approach relies on recorded history and data mining methods, such as the C4.5 classification algorithm. As a result, the approach modifies decision strategies in a particular way, by adding a single predictive rule in each iteration. This leaves the question whether there are other strategies (and ways of creating them) that may be as or more successful than the strategies from Studies 1 and 2.

Study 2 focused on the performance of decision strategies with increased feedforward control and improved mental models, as was suggested by results from Study 1. Study 3 investigates the previously posed question based on the success of some very simple strategies in Study 1. A prioritization strategy with a minimal mental model (treating conditions serially in a specific order, e.g., LGB or LBG) was shown to be robust and obtain near-optimal performance in a variety of settings in Study 1. Even the random choice strategy results in significantly better treatment than some of the principled treatment strategies, such as closest-to-goal. The success of these weak strategies can be explained in terms of their environment. In the prioritization of care

problem, different prioritization moves have different levels of power to reduce risk. For the investigated populations, a single lipids move generally lowers cardiovascular risk more than a single glucose move, which generally lowers risk more than a single blood pressure move. As a result, a feedback strategy that checks whether a condition is above evidence-based goals and follows this same ordering can achieve close to optimal performance. Even though the strategy uses feedback control and a minimal mental model, it is *ecologically rational* and exploits the structure of its environment to succeed (Gigerenzer 2001). Such feedback strategies may be effective on average but fail for particular subgroups, especially complex or hard-to-treat patients (such as those with high blood pressure and glucose even though they are already on several medications). Failures on these difficult cases may not significantly impact the strategy's average performance, because most patients do not fall into this subgroup; however, especially in healthcare, the N=1 case is of interest. So, we may ask, what kind of strategy would be well-adapted to an environment featuring difficult cases? By regarding the performance of decision strategies as a problem of adaptation, we can consider strategies as evolved structures. The question then becomes whether we can develop a process which creates decision strategies that achieve a desired level of performance. This is the focus of Study 3.

Study 3 develops an evolutionary approach to creating decision strategies. This approach was implemented using the simulation platform described in Section 6 to evolve decision strategies for the prioritization of care problem. In Study 3, versions of strategies that are comparable to Studies 1 and 2 have been evolved. Heuristic strategies,

such as treating in sequence, tend to evolve early and frequently, supporting Gigerenzer's ecological rationality argument (Gigerenzer & Todd 2012). Heuristic strategies tend to be robust enough to obtain sufficient fitness and are consequently found throughout future generations. On the other hand, strategies with greater numbers of rules often have greater fitness than simpler strategies in one generation, but are more sensitive to changes in patient population (because the population used for fitness evaluation changes from generation to generation) and can be selected against in the next few generations. This ties into Heiner's (1983) argument that uncertainty in the environment leads to predictability in action. Specifically, Heiner proposes that agents' inability to fully decipher the complexity of decision problems prevents them from selecting the most preferable alternative in many cases. Instead, in the face of uncertainty, agents revert to a smaller set of reliable, i.e., robust, behaviors, which in turn leads to more predictable responses (Heiner 1983). Therefore, strategies that evolve to contain a large number of rules at the expense of their robustness do not have a long-term evolutionary advantage because the strategy becomes unreliable. The ability to decipher the complexity of the environment (in Heiner's terms) is a function of the mental model. An accurate mental model enables effective feedforward control, as demonstrated in Study 1. However, if the environment changes in ways that are not represented in the mental model, the feedforward strategy lacks adaptability.

Study 3 also found several context-specific results. In the cost-effectiveness context, strategies evolved which focused only on the treatment of subgroups of patients with the greatest reversible risk, such as older patients and males. This type of discrimination

provided an evolutionary advantage for these strategies because neglecting treatments for younger patients and females resulted in cost savings. In other words, discrimination can be an adaptive response to the incentive structure of the environment, especially if the cost of actions is a fitness consideration. As seen in Study 3, the result can be unintended consequences, such as patients not receiving treatment. In the clinical domain, this is disconcerting because there already exists a predisposition to commit errors of omission, i.e., failures to treat patients who are not at evidence-based goals (Ramsey 2010). These errors are due to medical training, which emphasizes "first, do no harm"; lack of training and experience with chronically ill patient cases; and bounded rationality (Ramsey 2010). The research presented here indicates that cost-related incentives could create an environment that reinforces such errors.

The most challenging problem for evolved strategies was treating patients with varying adherence. Only two types of strategies were found to be successful in adherence management: feedforward strategies that examine expectation discrepancies, and feedback strategies that prescribed adherence treatments to all patients early on. In the latter case, the strategy uses a heuristic to treat everyone for adherence. Another interesting observation from Study 3 is that strategies that evolved to use more aggressive scheduling – thereby creating more treatment opportunities in the one-year timeframe – focused more on glucose and blood pressure treatment than strategies that treated less frequently which concentrated more on lipids treatment. This suggests a novel approach to prioritization based on the expectation of how often a patient can be treated. If the goal is risk reduction and it is unlikely that a patient will be seen more than two or three times

a year, initiating lipids treatment would be preferred. However, if the same patient can be seen more than seven times in a given year, the risk benefit of initiating and adjusting glucose or blood pressure treatment could be greater than that of lipids treatment after one year of treatment.

The cumulative results of studies 1, 2, and 3 show that decision strategies can be improved through learning methods, which use past experience to increase the accuracy of mental models, or through adaptation to their environments, which enables strategies to use the environment as scaffolding (Clark 1998).

7.1. Limitations

The research presented here is subject to several limitations. The results were obtained using simulation, namely the Simcare patient simulation model (Dutta et al. 2005) and the UKPDS Risk Engine (Stevens et al. 2001). Even though the simulation tools are sophisticated and have been subjected to extensive validation efforts, the simulation represents a model and is necessarily incomplete. However, the purpose of using simulation in the research presented here is to demonstrate and analyze decision strategies in a context that represents the real world, not to create medical guidelines or make treatment recommendations for specific clinical patients. The Simcare model has been suitably validated for this purpose, both against real-world randomized clinical trials and N=1 studies of individual patients under clinical care (McCabe et al. 2010; McCabe 2012). Moreover, the Simcare model has been used to train physicians as well as residents in the treatment of type 2 diabetes patients (Sperl-Hillen et al. 2010).

The UKPDS Risk Engine, used in this research, was developed based on findings and observations from the UK Prospective Diabetes Study, a randomized clinical trial that took place at 23 sites in the United Kingdom between 1977 and 1997 and involved 5,102 type 2 diabetes patients. Therefore, a concern is that the risk engine is outdated and based on care in the United Kingdom which had a public healthcare system at the time of the study. In addition, the Risk Engine does not take into account some risk factors which may be of interest, such as body mass index (BMI) or aspirin use. However, the analysis from the studies presented here can be readily repeated using a different risk model as long it produces the same types of outputs, i.e., probabilities of experiencing an adverse event in a given timeframe for an individual patient.

A further limitation is that the results presented here are based on a single population under care at a regional health system in the United States. The populations under care at other health care providers may have different characteristics as considerable variation exists in the levels of diabetes care across states (Coffey et al. 2004). The population used for this study represents a health system with a quality of care level considerably higher than average. Replicating the experiments presented here on a patient population that experienced different care is desirable.

7.2. Future Work

This research was conducted in the context of chronic disease care, but its scope provides contributions to the dynamic decision making literature. One avenue of future work is to extend the presented work to other dynamic environments.

Within the clinical domain, future work includes incorporating additional risk factors of interest to the medical community (such as smoking cessation and aspirin use) and alternative risk models (for example, taking a Bayesian approach rather than a regression approach to risk estimation). The latter will help overcome the limitations of the UKPDS Risk Engine discussed above.

The evolutionary approach used to develop decision strategies in this research also shows great promise and provides a number of interesting directions for future work. One direction is to develop decision strategies by co-evolution of expectations and condition-action rules. The resulting strategies may have interesting and informative features, such as different expectations for different subgroups of patients, or different sets of expectations for high- and low-adherence patients.

Another interesting direction for future work is the use of more sophisticated fitness functions. Especially in the clinical domain, the outcome for an individual patient is of great importance. A fitness function could judge the performance of a strategy overall based on a percentage of the worst-off patients, i.e., those with the highest risk after treatment. This would favor the evolution of strategies that improve treatment for the worst-off patients and lower the maximum risk of the population rather than the mean.

Finally, an interesting future direction is the evolution of *non-deterministic decision strategies*, building on McCarthy's (1963) ambiguous functions. Such strategies would have a built-in mechanism to allow for resulting actions to be rejected. In this case, the strategy would simply backtrack and make another viable suggestion. For example, in the

clinical domain, patients may reject treatment options such as smoking cessation. In this case, a non-deterministic decision strategy would be able to produce the next best alternative for risk reduction automatically. In this way, it becomes possible to evolve decision strategies whose fitness takes into account the effects of patient preferences and choices, which are unknown at the time of strategy creation.

7.3. Conclusions

The research presented here contributes to dynamic decision making by focusing on decision strategies rather than individual approaches to solving problems in dynamic environments, specifically in the context of chronic disease care. Meehl (1996) argued for the use of "mechanical prediction tools", of which decision strategies are an instance, rather than individual clinical judgment. Grove et al. (2000) found empirically that mechanical tools consistently outperform human decision makers in a variety of clinical tasks and settings. This dissertation investigates decision strategies and develops approaches of constructing and modifying strategies for a given environment using machine learning and genetic programming.

In all three studies, it was found that effective decision strategies require a certain level of complexity; however, this complexity results in a trade-off with robustness. In Study 1, it was shown that the performance of a very effective feedforward strategy can degrade considerably when its mental model is changed only slightly. However, as shown in Studies 2 and 3, additional complexity in the form of feedforward thinking and its associated stronger mental model is required to achieve satisfactory treatment outcomes

for some patients, such as patients who suffer from multiple conditions that are far from evidence-based goals, or patients with low or varying adherence.

Newell (1969) proposed that problem solving methods can be plotted on a graph that shows a trade-off between power and generality, as shown in Figure 48. The same appears to be true for decision strategies in terms of power versus robustness. In general, a one-size-fits-all treatment strategy will not perform as well for a subgroup of patients as a strategy designed especially for this group. However, as seen in the case of adherence, it is not always easy to identify whether a patient belongs to a certain subgroup.

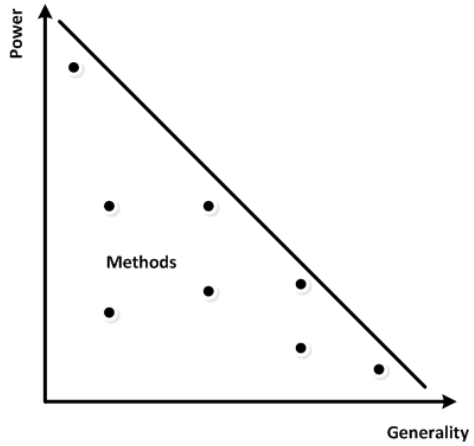

Figure 48. Generality vs. power, based on Newell (1969)

Especially in medicine, we are ultimately looking for the decision strategy that produces the best outcome for the individual case. However, since the knowledge to treat patients is always a generalization of some form (from experience, clinical trials, guidelines, etc.), we need to be concerned with the robustness of this knowledge. If we are wrong about the type of patient, may we do more harm than if the patient was just

treated with a one-size-fits-all strategy (Veazie et al. 2009)? As Heiner (1983) points out, in the presence of uncertainty, predictable behavior emerges. If we cannot reliably assign a patient to a category, we must fall back on a limited number of strategies that will work even if the assignment of a patient to a category is uncertain. The question becomes, then, "with what strategies should we populate this limited space?" The studies presented here provide insights about how to create candidate strategies to answer this question by means of evolution or machine learning.

This research suggests the following questions for future work: (1) What is the nature of the decision strategies at the power vs. robustness frontier? (2) What is the role of the environment in determining the frontier? (3) Can the frontier be changed, and if so, how? (4) How can strategies be constructed at a given frontier?

References

ACCORD Study Group, 2008. Effects of Intensive Glucose Lowering in Type 2 Diabetes. *New England Journal of Medicine*, 358(24), pp.2545–2559.

van der Aalst, W.M.P. & Weijters, A.J.M.M., 2004. Process mining: a research agenda. *Computers in Industry*, 53(3), pp.231–244.

American Diabetes Association, 2007. National Diabetes Fact Sheet. Available at: http://www.diabetes.org/diabetes-basics/diabetes-statistics/.

American Diabetes Association, 2012. Standards of medical care in diabetes--2012. *Diabetes Care*, 35 Suppl 1(October 2011), pp.S11–63.

Ashby, W.R., 1968. Variety, Constraint, and the Law of Requisite Variety. In W. Buckley, ed. *Modern Systems Research for the Behavioral Scientist*. Chicago, IL: Adline Press.

Atkins, P.W.B., Wood, R.E. & Rutgers, P.J., 2002. The effects of feedback format on dynamic decision making. *Organizational Behavior and Human Decision Processes*, 88, pp.587–604.

Avins, A.L., 2010. When Clinical Practice Guidelines Meet the Black Box. *Archives of Internal Medicine*, 170(12), pp.1013–1014.

Bainbridge, L., 1981. Mathematical equations or processing routines? In J. Rasmussen & W. B. Rouse, eds. *Human Detection and Diagnosis of Systems Failures*. New York, NY: Plenum.

Banzhaf, W. et al., 1998. *Genetic Programming: An Introduction*, San Francisco, CA: Morgan Kaufmann.

Beach, L.R. & Mitchell, T.R., 1978. A Contingency Model for the Selection of Decision Strategies. *The Academy of Management Review*, 3(3), pp.439–449.

Bell, C.M. et al., 2006. Bias in published cost effectiveness studies: systematic review. *British Medical Journal*, 332, p.699.

Bland, M., 1995. *An Introduction to Medical Statistics*, Oxford University Press.

Blickle, T. & Thiele, L., 1996. A Comparison of Selection Schemes used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4), pp.361–394.

Brehmer, B., 1992. Dynamic decision making: Human control of complex systems. *Acta Psychologica*, 81, pp.211–241.

Brehmer, B., 1990. Strategies in Real-Time, Dynamic Decision Making. In *Insights in Decision Making*. Chicago, IL: The University of Chicago Press.

Briss, P.A. et al., 2000. Developing an Evidence-Based Guide to Community Preventive Services — Methods. *American Journal of Preventive Medicine*, 18(99), pp.35–43.

Broadbent, D.E., FitzGerald, P. & Broadbent, M.H.P., 1986. Implicit and explicit knowledge in the control of complex systems. *British Journal of Psychology*, 77(1), pp.33–50.

Bäck, T., 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press.

Cai, W. et al., 2006. Heat transfer correlations by symbolic regression. *International Journal of Heat and Mass Transfer*, 49(23-24), pp.4352–4359.

Chalmers, J. & Cooper, M.E., 2008. UKPDS and the Legacy Effect. *New England Journal of Medicine*, 359(15), pp.1618–1620.

Chapman, R.H. et al., 2000. Comprehensive League Table of Cost-Utility Ratios and a Sub-table of ``Panel-worthy'' Studies. *Medical Decision Making*, 20, pp.451–458.

Chen, S.-H. & Liao, C.-C., 2005. Agent-based computational modeling of the stock price-volume relation. *Information Sciences*, 170(1), pp.75–100. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0020025503004365.

Clark, A., 1998. *Being There: Putting Brain, Body and World Together Again*, Cambridge, MA: The MIT Press.

Clarke, P.M. et al., 2004. A model to estimate the lifetime health outcomes of patients with type 2 diabetes: the United Kingdom Prospective Diabetes Study (UKPDS) Outcomes Model. *Diabetologia*, 47(10), pp.1747–1759.

Coffey, R.M., Matthews, T.L. & Kelly, M., 2004. *Diabetes Care Quality Improvement: A Resource Guide for State Action*, Available at: http://www.ahrq.gov/qual/diabqual/diabqguidemod1a.htm.

Conant, R.C. & Ashby, W.R., 1970. Every Good Regulator of a System Must Be a Model of that System. *International Journal of Systems Sciences*, 1(2), pp.89–97.

Denton, B.T. et al., 2009. Optimizing the start time of statin therapy for patients with diabetes. *Medical Decision Making*, 29(3), pp.351–67.

Deschaine, L., 2006. Using Information fusion, machine learning, and global optimisation to increase the accuracy of finding and understanding items interest in the subsurface. *geoDrilling International*, (122), pp.30–32. Available at: http://www.mining-journal.com/gdi_magazine/pdf/GDI0605scr.pdf.

Devereaux, P.J. & Yusuf, S., 2003. The evolution of the randomized controlled trial and its role in evidence-based decision making. *Journal of Internal Medicine*, 254(2), pp.105–13.

Doyle, J.K. & Ford, D.N., 1998. Mental models concepts for system dynamics research. *System Dynamics Review*, 14, pp.3–29.

Dutta, P. et al., 2005. SimCare: A Model for Studying Physician Decisionmaking Activity. In *Advances in Patient Safety: From Research to Implementation*. AHRQ, pp. 179–192.

D'Agostino, R.B. et al., 2008. General Cardiovascular Risk Profile for Use in Primary Care: The Framingham Heart Study. *Circulation*, 117, pp.743–753.

Edwards, W., 1962. Dynamic decision theory and probabilistic information processing. *Human Factors*, 4(2), pp.59–73.

Forrester, J., 1971. Counterintuitive Behavior of Social Systems. *Technology Review*, 73(3), pp.52–68.

Forster, M.R., 2000. Key Concepts in Model Selection: Performance and Generalizability. *Journal of Mathematical Psychology*, 44(1), pp.205–231.

Friedman, J.H., 2001. The Role of Statistics in the Data Revolution? *International Statistical Review*, 69(1), pp.5–10.

Funes, P. et al., 1998. Animal-animat coevolution: Using the animal population as fitness function. In R. Pfeifer et al., eds. *Proceedings of the International Conference on Simulation of Adaptive Behavior*. MIT Press, pp. 525–533. Available at: http://www.demo.cs.brandeis.edu/papers/tronsab98.ps.

Funke, J., 1991. Solving Complex Problems: Exploration and Control of Complex Systems. In R. Sternberg & P. Frensch, eds. *Complex Problem Solving - Principles and Mechanisms*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 185–222.

Garcia, C.E. & Morari, M., 1982. Internal model control. A unifying review and some new results. *Industrial & Engineering Chemistry Process Design and Development*, 21(2), pp.308–323.

Gibson, F.P., Fichman, M. & Plaut, D.C., 1997. Learning in Dynamic Decision Tasks: Computational Model and Empirical Evidence. *Organizational Behavior and Human Decision Processes*, 71(1), pp.1–35.

Gigerenzer, G., 2001. *Bounded Rationality: The Adaptive Toolbox* G. Gigerenzer & R. Selten, eds., The MIT Press.

Gigerenzer, G. & Todd, P.M., 2012. *Ecological Rationality: Intelligence in the World*, Oxford University Press.

Gigerenzer, G. & Todd, P.M., 1999. Simple Heuristics That Make Us Smart. In New York, NY: Oxford University Press.

Gilmer, T.P. et al., 2012. Cost-Effectiveness of an Electronic Medical Record Based Clinical Decision Support System. *Health Services Research*, pp.1–22.

Gilmer, T.P. et al., 2006. Impact of office systems and improvement strategies on costs of care for adults with diabetes. *Diabetes Care*, 29(6), pp.1242–8.

Gilmer, T.P. et al., 2005. Predictors of Health Care Costs in Adults. *Diabetes Care*, 28(1), pp.59–64.

Gode, D.K. & Sunder, S., 1993. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economics*, 101(1), pp.119–138.

Gold, M.R. et al., 1996. *Cost-Effectiveness in Health and Medicine*, New York, NY: Oxford University Press.

Gonzalez, C., Lerch, J.F. & Lebiere, C., 2003. Instance-based learning in dynamic decision making. *Cognitive Science*, 27, pp.591–635.

Gonzalez, C., Vanyukov, P. & Martin, M.K., 2005. The use of microworlds to study dynamic decision making. *Computers in Human Behavior*, 21, pp.273–286.

Grove, W.M. et al., 2000. Clinical versus mechanical prediction: A meta-analysis. *Psychological Assessment*, 12(1), pp.19–30.

Gustafson, S., Burke, E.K. & Krasnogor, N., 2005. On Improving Genetic Programming for Symbolic Regression. In *Proceedings of the Congress on Evolutionary Computation*. Ieee, pp. 912–919.

Hauptman, A. & Sipper, M., 2007. Evolution of an Efficient Search Algorithm for the Mate-In-N Problem in Chess. In M. Ebner et al., eds. *Proceedings of the European Conference on Genetic Programming*. Springer, pp. 78–89.

Heiner, R.A., 1983. The Origin of Predictable Behavior. *American Economic Review*, 73(4), pp.560–595.

Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Bradford MIT Press.

Institute for Clinical Systems Improvement, 2009. *Health Care Guideline: Diagnosis and Management of Type 2 Diabetes Mellitus in Adults*, Minneapolis, MN.

International Organization for Standardization, 1996. ISO/IEC 14977:1996 Information technology -- Syntactic metalanguage -- Extended BNF.

Jaen, C.R., Stange, K.C. & Nutting, P.A., 1994. Competing demands of primary care: A model for the delivery of clinical preventive services. *Journal of Family Practice*, 38(2), pp.166–171.

Johnson, B., 2010. *Algorithmic Trading and DMA: An introduction to direct access trading strategies*, 4Myeloma Press.

Kattan, A. & Poli, R., 2008. Evolutionary lossless compression with GP-ZIP. In J. Wang, ed. *Proceedings of the Conference on Genetic and Evolutionary Computation*. IEEE Press, pp. 1211–1218.

Keijzer, M., 2004. Scaled Symbolic Regression. *Genetic Programming and Evolvable Machines*, 5(3), pp.259–269.

Koehler, E., Brown, E. & Haneuse, S.J.-P. a, 2009. On the Assessment of Monte Carlo Error in Simulation-Based Statistical Analyses. *The American Statistician*, 63(2), pp.155–162.

Koza, J.R. et al., 1999. *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann.

Koza, J.R. et al., 2003. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Springer.

Koza, J.R., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press.

Langdon, W.B. & Poli, R., 2002. *Foundations of Genetic Programming*,

Lerch, J.F. & Harter, D.E., 2001. Cognitive Support for Real-Time Dynamic Decision Making. *Information Systems Research*, 12(1), pp.63–82.

Mackinnon, A.J. & Wearing, A.J., 1985. Systems analysis and dynamic decision making. *Acta Psychologica*, 58, pp.159–172.

Marr, D.C., 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, San Francisco, CA: W. H. Freeman.

Mazze, R.S. et al., 2012. *Staged Diabetes Management*, Wiley-Blackwell.

Mazze, R.S. et al., 2007. *Staged Diabetes Management: A Systematic Approach*, Wiley.

Mazze, R.S. et al., 2005. *Staged Diabetes Management: Prevention, Detection and Treatment of Diabetes in Adults Quick Guide*, Minneapolis, MN: Matrex.

McCabe, R.M., 2012. *Coherence and correspondence validation of the SimCare patient model*. University of Minnesota.

McCabe, R.M. et al., 2010. Validation of the SimCare Model: A Computational Model of Individual Patients with Type 2 Diabetes. *Diabetes*, 59 Suppl 1, p.A353.

McCarthy, J., 1963. A basis for a mathematical theory of computation. In *Computer Programming and Formal Systems*. North-Holland, pp. 33–70.

Meehl, P.E., 1996. *Clinical versus Statistical Prediction: A Theoretical Analysis and a Review of the Evidence*, Jason Aronson.

Miller, B.L. & Goldberg, D.E., 1995. Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*, 9(95006), pp.193–212.

Mitchell, T.M., 1997. *Machine Learning*, McGraw-Hill Science/Engineering/Math.

Neely, C.J., Weller, P.A. & Ulrich, J.M., 2009. The Adaptive Markets Hypothesis: Evidence from the Foreign Exchange Market. *Journal of Financial and Quantitative Analysis*, 44(02), p.467. Available at: http://www.journals.cambridge.org/abstract_S0022109009090103.

Newell, A., 1990. Foundations of Cognitive Science. Unified Theories of Cognition. In Cambridge, NY: Harvard University Press.

Newell, A., 1969. Heuristic Programming: Ill-Structured Problems. In J. S. Aronofsky, ed. *Progress in Operations Research*. Wiley, pp. 3–54.

Newell, A. & Ernst, G.W., 1969. *GPS : A Case Study in Generality and Problem Solving*, New York, NY: Academic Press.

Newell, A., Shaw, J.C. & Simon, H.A., 1959. Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*. UNESCO House, pp. 256–264.

Nocedal, J. & Wright, S.J., 2006. *Numerical Optimization (Second Edition)*, Springer.

Oremland, R.S., Stolz, J.F. & Hollibaugh, J.T., 2004. The microbial arsenic cycle in Mono Lake, California. *FEMS Microbiology Ecology*, 48(1), pp.15–27.

O'Brien, J.A. et al., 1998. Direct Medical Costs of Complications Resulting from Type 2 Diabetes in the U.S. *Diabetes Care*, 21(7), pp.1122–1128.

O'Brien, J.A., Patrick, A.R. & Caro, J., 2003. Estimates of Direct Medical Costs for Microvascular and Macrovascular Complications Resulting from Type 2 Diabetes Mellitus in the United States in 2000. *Clinical Therapeutics*, 25(3), pp.1017–1038.

PDR Staff, 2011. *Physicians' Desk Reference, 66th Edition*, PDR Network.

Parchman, M.L. et al., 2007. Competing Demands or Clinical Inertia: The Case of Elevated Glycosylated Hemoglobin. *Annals of Family Medicine*, 5(3), pp.196–201.

Phillips, L.S. et al., 2001. Clinical Inertia. *Ann Intern Med*, 135, pp.825–834.

Piette, J.D. & Kerr, E. a, 2006. The impact of comorbid chronic conditions on diabetes care. *Diabetes Care*, 29(3), pp.725–31.

Quinlan, J.R., 1993. *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann.

Quinlan, J.R., 1987. Generating production rules from decision trees. In *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 304–307.

Ramsey, G.W., 2010. *Success and Failure in Dynamic Decision Environments: Understanding Treatment Strategies for Patients with a Chronic Disease*. University of Minnesota.

Ramsey, G.W. et al., 2010. Using Functional Data Analysis to Identify Physician Decision Strategies which Lead to Better Type 2 Diabetes Patient Outcomes. In *Proceedings of the ACM International Conference on Health Informatics*.

Rapoport, A., 1975. Research paradigms for the study of dynamic decision behavior. In D. Wendt & C. Vlek, eds. *Utility, Probability and Human Decision Making*. Dortrecht, Netherlands: Reidel.

Riethof, M. et al., 2012. *Diagnosis and Management of Type 2 Diabetes Mellitus in Adults*,

Sadoyan, H., Zakarian, A. & Mohanty, P., 2004. Data Mining Algorithm for Manufacturing Process Control. *Interational Journal of Advanced Manufacturing Technology*, 28, pp.342–350.

Sassi, F., 2006. Calculating QALYs, comparing QALY and DALY calculations. *Health Policy and Planning*, 21(5), pp.402–408.

Seborg, D.E., Edgar, T.F. & Mellichamp, D., 2004. *Process Dynamics & Control*, New York, NY: Wiley.

Seborg, D.E., Edgar, T.F. & Shah, S.L., 1986. Adaptive Control Strategies for Process Control: A Survey. *American Institute of Chemical Engineers Journal*, 32(6), pp.881–913.

Selby, J.V. et al., 2003. Determining the value of disease management programs. *Joint Commission Journal on Quality and Patient Safety*, 29(9), pp.491–499.

Shah, B.R. et al., 2005. Clinical inertia in response to inadequate glycemic control. *Diabetes Care*, 28(3), p.600.

Simon, H.A., 1969. *The Sciences of the Artificial*, Cambridge, MA: The MIT Press.

Spector, L. et al., 1999. Finding a better-than-classical quantum AND/OR algorithm using genetic programming. In *Proceedings of the Congress on Evolutionary Computation*. Ieee, pp. 2239–2246.

Sperl-Hillen, J.M. et al., 2010. Simulated Physician Learning Program Improves Glucose Control in Adults With Diabetes. *Diabetes Care*, 33(8), pp.1727–1733.

Sterman, J.D., 1989. Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes*, 43, pp.301–335.

Stevens, R.J. et al., 2001. The UKPDS risk engine: a model for the risk of coronary heart disease in Type II diabetes (UKPDS 56). *Clinical Science*, 101, pp.671–679.

Stratton, I.M. et al., 2000. Association of glycaemia with macrovascular and microvascular complications of type 2 diabetes (UKPDS 35): prospective observational study. *British Medical Journal*, 321, pp.405–412.

Tan, P.-N., Steinbach, M. & Kumar, V., 2006. *Introduction to Data Mining*, Addison-Wesley Publishing Company.

Turner, R.C. et al., 1991. UK Prospective Diabetes Study (UKPDS): Study design, progress and performance. *Diabetologia*, 34(12), pp.877–890.

Usman, I. et al., 2007. Image Authenticity and Perceptual Optimization via Genetic Algorithm and a Dependence Neighborhood. *International Journal of Applied Mathematics and Computer Sciences*, 4(1), pp.615–620. Available at: http://www.waset.org/ijamcs/v4/v4-1-7.pdf.

Veazie, P.J., Johnson, P.E. & O'Connor, P.J., 2009. Is there a downside to customizing care? Implications of general and patient-specific treatment strategies. *Journal of Evaluation in Clinical Practice*, 15(6), pp.1171–6.

Wagner, E.H. et al., 2001. Chronic care clinics for diabetes in primary care: a system-wide randomized trial. *Diabetes Care*, 24(4), pp.695–700.

Xie, H., Zhang, M. & Andreae, P., 2006. Genetic Programming for Automatic Stress Detection in Spoken English. In F. Rothlauf et al., eds. *Applications of Evolutionary Computing EvoWorkshops2006 EvoBIO EvoCOMNET EvoHOT EvoIASP EvoInteraction EvoMUSART EvoSTOC*. Springer Verlag, pp. 460–471. Available at: http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=3907&spage=460.

Yu, T., Chen, S.-H. & Kuo, T.-W., 2004. Discovering Financial Technical Trading Rules Using Genetic Programming with Lambda Abstraction. In U.-M. O'Reilly et al., eds. *Genetic Programming Theory and Practice II*. Springer, pp. 11–30.

Zhang, M. & Smart, W., 2006. Using Gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. *Pattern Recognition Letters*, 27(11), pp.1266–1274. Available at: http://www.mcs.vuw.ac.nz/comp/Publications/CS-TR-05-5.abs.html.

Zhao, H., 2007. A Multi-objective Genetic Programming Approach to Developing Pareto Optimal Decision Trees. *Decision Support Systems*, 43(3), pp.809–826.

Appendix A. From Risk to Events: Monte Carlo Simulation

The Simcare patient model does not directly simulate events. When events need to be simulated for experiments, they are generated as follows. First, the UKPDS Risk Engine is used to estimate the probability of that event (e.g., a heart attack) occurring for a given synthetic patient (based on the current patient state) (Stevens et al. 2001). Next, using a Monte Carlo method, an event is generated by creating a random number r (using a random number generator) from the standard uniform distribution and comparing it to the probability p of the event. If $r < p$, the synthetic patient is said to have experienced the event in this run.

This process is repeated for each patient in a group to simulate events for an entire group and obtaining a discrete number of events. When using Monte Carlo methods, runs are repeated many times (in the case of these studies, 1,000 repetitions) using a different seed for the random number generator each time. This allows estimation of Monte Carlo error to validate that a given result is not due to a particular series of randomly generated numbers but rather a robust finding (Koehler et al. 2009). This is consistent with the method used to create simulated events for the validation of the Simcare patient model against randomized clinical trials (McCabe 2012).

Appendix B. Sensitivity Analysis for Study 1

Sensitivity Analysis: Population

To verify the robustness of our findings for specific subpopulations, we selected a subgroup for which prioritization is even more critical. We selected the subgroup of patients which were in the 80th or higher percentile for all three conditions, ensuring that prioritization and multiple treatment moves for all three conditions are required. The characteristics for this sub-population are listed in Table 25 and indicate that this subpopulation exhibits higher values for most risk factors.

Table 25. Sub-population characteristics (N=667)

Characteristic	Mean	SD
Age	58.9	13.3
Female (%)	48.3	
BMI (kg/m^2)	35.7	9.2
Hemoglobin A1c (%)	9.7	1.4
Systolic blood pressure (mmHg)	152.6	11.6
Low-density lipoprotein LDL (mg/dL)	154.5	28.6
High-density lipoprotein HDL (mg/dL)	43.3	13.0
Triglycerides (mg/dL)	198.5	118.8
Current smoker (%)	6	

The event rate resulting from the application of all prioritization strategies are tabulated in Table 26 below.

Table 26. 10-year cardiovascular event rate for base case population and selected sub-population

Strategy	10 year event rate after 4 moves (entire population) N = 4,734	10 year event rate after 4 moves (sub-population) N = 667
No additional treatment	.320 −	.375 −
Risk-based	.259 *	.267 +*
H1: Random Choice	.294	.303
H2a: Serial BGL	.320 −	.350 −
H2b: Serial BLG	.298	.343 −
H2c: Serial GBL	.319 −	.321
H2d: Serial GLB	.313	.316
H2e: Serial LBG	.264 +*	.277 +*
H2f: Serial LGB	.258 +*	.266 +*
H3a: Closest-to-goal	.316 −	.340 −
H3b: Farthest-from-goal	.286	.292
H3c: Closest-to-goal (pc)	.312	.314
H3d: Farthest-from-goal (pc)	.302	.316
H3e: Closest-to-goal (ex)	.308	.311
H3f: Farthest-from-goal (ex)	.313	.324 −
Optimal (post hoc)	.258 +*	.266 *

+ significantly lower than random choice, − significantly higher than random choice
* = not significantly different from optimal

The findings from the base case are not only robust but also amplified for this sub-population with an increased need for prioritization. Risk-based prioritization as well as serial treatment strategies that focus on lipids first still reach performance that is not significantly different from what could be reached with an optimal strategy. The serial treatment BLG strategy performs significantly worse for the subgroup, and serial BGL had a 6.2% higher event rate for the full population but has a 8.4% higher event rate for the subgroup. Risk-based prioritization reduces the event rate by 6.1% from "no additional treatment" in the base case, but by 10.8% for the subpopulation.

Sensitivity Analysis: Number of Visits

Table 27. 10-year cardiovascular event rate for prioritization strategies after 4 and 8 treatment moves with 90-day intervals

Strategy	10 year event rate after 4 moves	10 year event rate after 8 moves
No additional treatment	.320 −	.334 −
Risk-based	.259 *	.220 *
H1: Random Choice	.294	.231 *
H2a: Serial BGL	.320 −	.288 −
H2b: Serial BLG	.298	.250
H2c: Serial GBL	.319 −	.270 −
H2d: Serial GLB	.313	.254 −
H2e: Serial LBG	.264 +*	.233 *
H2f: Serial LGB	.258 +*	.219 *
H3a: Closest-to-goal (%)	.316 −	.268 −
H3b: Farthest-from-goal (%)	.286	.236 *
H3c: Closest-to-goal (pc)	.312	.262 −
H3d: Farthest-from-goal (pc)	.302	.264 −
H3e: Closest-to-goal (ex)	.308	.262 −
H3f: Farthest-from-goal (ex)	.313	.261 −
Optimal (post hoc)	.258 +*	.219 *

+ significantly lower than random choice, − significantly higher than random choice
* = not significantly different from optimal

After two years of treatment (eight visits with treatment intensification), most patients reach goals or converge close to goal, as a result, risk-based prioritization, farthest-from-goal, and random choice all achieve outcomes that are not significantly different from optimal prioritization. This convergence is encouraging as a demonstration that getting patients into treatment and repeated intensification to bring patients to goal results in significant risk reductions. This point is especially well illustrated by the performance of random choice – independent of prioritization, eight visits with intensification are sufficient for the examined population to lower event rate to the point where it is no longer different from an optimal prioritization strategy. However, even after eight visits, several strategies still perform worse than random choice, specifically closest-to-goal and serial treatment strategies that postpone treatment of lipids. This

underlines the importance of examining and designing appropriate prioritization strategies to guide early treatment decisions.

Sensitivity Analysis: Visit Interval

The assumption of four visits in a year to focus on chronic conditions may be optimistic. Therefore, we conducted sensitivity analysis to assess the impact of lowering this number to two visits per year and one visit per year.

Table 28. 10-year cardiovascular event rate for prioritization strategies after 4 treatment moves with 90-, 180-, and 360-day intervals

Strategy	10 year event rate after 4 moves (every 90 days)	10 year event rate after 4 moves (every 180 days)	10 year event rate after 4 moves (every 360 days)
Study timeframe	1 year	2 years	4 years
No additional treatment	.320 −	.353 −	.406 −
Risk-based	.259 *	.289 +*	.304 +*
H1: Random Choice	.294	.319	.337
H2a: Serial BGL	.320 −	.342 −	.386 −
H2b: Serial BLG	.298	.336	.381 −
H2c: Serial GBL	.319 −	.341 −	.350
H2d: Serial GLB	.313	.336	.345
H2e: Serial LBG	.264 +*	.300 *	.316 *
H2f: Serial LGB	.258 +*	.289 +*	.304 +*
H3a: Closest-to-goal (%)	.316 −	.363 −	.380 −
H3b: Farthest-from-goal (%)	.286	.314	.324 *
H3c: Closest-to-goal (pc)	.312	.336	.346
H3d: Farthest-from-goal (pc)	.302	.336	.351
H3e: Closest-to-goal (ex)	.308	.334	.353
H3f: Farthest-from-goal (ex)	.313	.348 −	.358
Optimal (post hoc)	.258 +*	.287 +*	.302 +*

+ significantly lower than random choice, − significantly higher than random choice

* = not significantly different from optimal

The findings from the base case are again shown to be robust. The four year case demonstrates that appropriate prioritization strategies can effectively counteract the risk-increasing effects of age and disease progression even with as little as one treatment move per year. Conversely, strategies that focus on moves that do not significantly

reduce the risk burden result in considerably higher event rates. The difference between risk-based choice and serial BGL (the worst performing strategy) was 6.1% with 90 day intervals but is 8.4% with 360 day intervals, corresponding to 118 additional events over ten years for the study population.

Appendix C. Genetic Programming Implementation Details for Study 3

Section 6 describes the approach to represent decision strategies so that genetic programming can be applied to evolve them. This Appendix describes the implementation details.

In genetic programming, programs consist of *terminals* (variables and constants) and *functions* (Koza 1992). For example, suppose the objective is to evolve a program that compares two Boolean (true or false) values to test if they are same. Figure 49 is a tree-representation of such a program. It tests whether two Boolean variables $\boldsymbol{D0}$ and $\boldsymbol{D1}$ have the same value, i.e., the program returns True if $\boldsymbol{D0}$ and $\boldsymbol{D1}$ both are True or both are False. The shaded nodes $\boldsymbol{D0}$ and $\boldsymbol{D1}$ are terminals; OR, AND, NOT are functions.

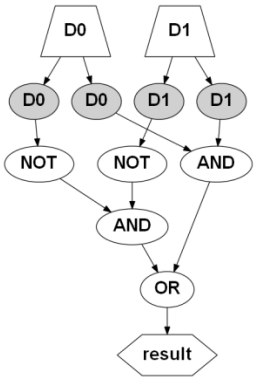

Figure 49. Representation of a Boolean even 2-parity function (Koza 1992)

Koza's (1992) method requires that terminals and functions satisfy the *closure* property, i.e., that any outputs can be used again as inputs. However, in the realm of

decision strategies, the functions to evolve are condition-action rule-sets: given a state as input, the evolved function is to output an action. The output of a decision strategy cannot directly be used as its input; the closure property is violated. Below, a grammatical form is introduced to describe decision strategies that permits their evolution without satisfying the closure property.

Structure of Decision Strategies: Context Free Grammar

The structure and constraints for a decision strategy are represented below as a context-free grammar in Extended Backus-Naur Form in Table 29 (International Organization for Standardization 1996).

Table 29. Extended Backus-Naur Form (EBNF) representation of decision strategies

1	strategy := action
2	action := action-primitive | action-function | rule
3	rule := predicate action action
4	predicate := expression-to-evaluate expression-to-evaluate
5	expression-to-evaluate := state-function | expectation-function | literal
6	state-function := state-function-primitive (expression-to-evaluate | state-var-primitive) (expression-to-evaluate | state-var-primitive)?
7	expectation-function := expectation-primitive (expectation-function | state-var-primitive) (action-function | action-primitive)
8	action-function := action-function-primitive (action-function | action-primitive)?

A *strategy* consists of one action, the top level node in a tree representation. If this action is an *action primitive*, the strategy is said to be trivial and has "one-size-fits-all" behavior. *Action primitives* are terminals. The typical case is that the top level node is a *rule*.

An *action* may be an *action primitive*, a *function returning an action*, or a *rule*.

A *rule* consists of a *predicate*, an *action* to be taken if the *predicate* is true (true branch), and an *action* to take when the *predicate* is false (false branch). A *rule* can be represented graphically as shown in Figure 50.

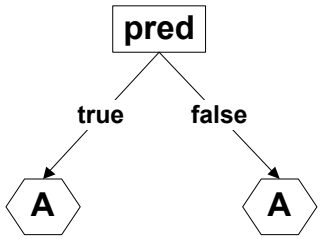

Figure 50. Graphical representation of a rule (pred = predicate, A = action)

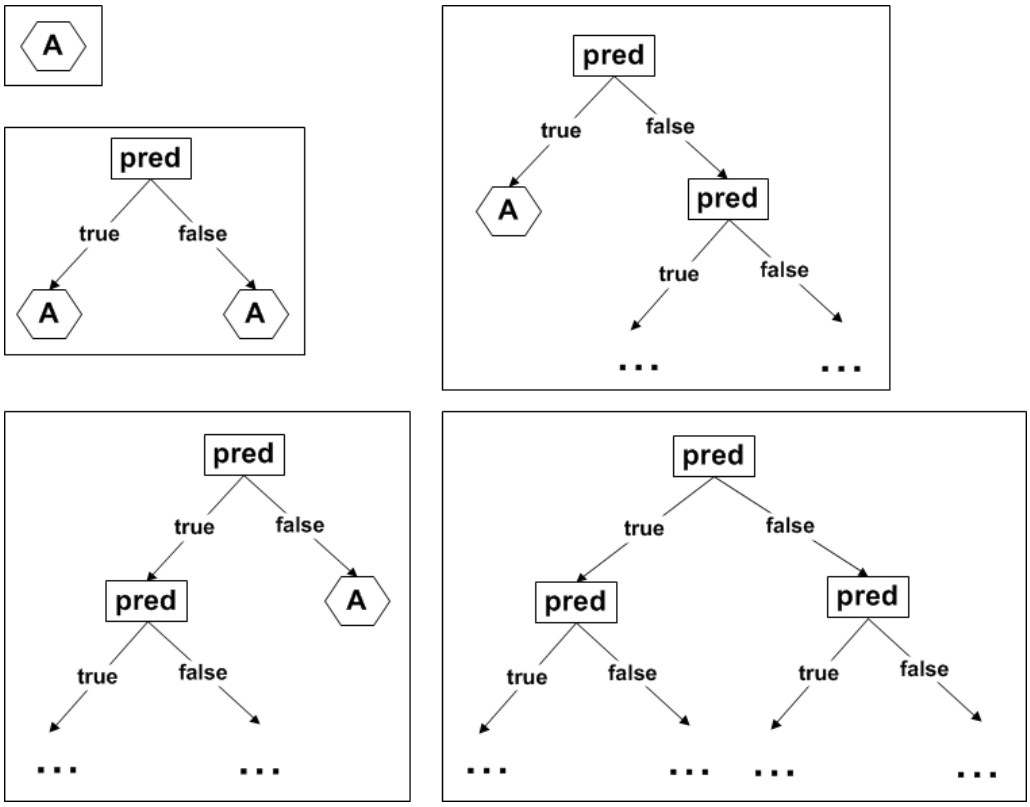

Figure 51. Graphical representation of five strategies satisfying the grammar

Figure 51 illustrates how the grammar rules 1-3 result in structures that can be represented in tree form.

A *predicate* consists of two *expressions to evaluate*, X and Y, and is always evaluated as $X > Y$?.

An *expression to evaluate* is either a *function returning a state* or a *literal*.

A *literal* is a random constant from the standard uniform distribution $U(0,1)$. The reason for this is given below when the evaluation of state variables is discussed.

A *function returning an action* consists of an *action function primitive* that takes one optional parameter. The parameter is either a *function returning an action* or an *action primitive*.

A *function returning a state* may be either a *state function* or an *expectation function*.

A *state function* consists of a *state function primitive* and takes one or two parameters, each of which may be either an *expression to evaluate* or a *state variable primitive*.

An *expectation function* consists of an *expectation primitive* and takes either a *state variable primitive* or a *state function* as first parameter, and either an *action primitive* or an *action function* as second parameter.

A *state function primitive* may take one or two parameters that are either *state variable primitives* or *expressions to evaluate*.

An *expectation primitive* takes two parameters, a *state variable primitive* and an *action primitive*, and returns the expected value of that state variable if that action had been taken.

A *state variable primitive* is represents a single state variable. The representation of state variables is discussed below.

Primitives

The provided grammar can be used to create decision strategies that can arbitrarily partition the patient space and assign actions to them. The limitations of the strategies are therefore determined by the available primitives. For example, without a state variable for blood pressure, a strategy that treats blood pressure effectively cannot evolve because blood pressure cannot be represented in the inverse model. The primitives used in Study 3 are described below.

The *state variable primitives* are listed in Table 30.

Table 30. State variable primitives

Variable Name	Description	Unit of Measure
EncDate	Encounter date in number of days since the first encounter with this patient	
EncNum	Number of times this patient has been seen	
Age	Patient's current age	years
Gender	Gender (0 = Male, 1 = Female)	
Afib	History of atrial fibrillation (0 = no, 1 = yes)	
Smoking	Smoking status (0 = Non-smoking, 1 = smoking)	
SBP	Systolic blood pressure	mmHg
SMBG	Self-monitored blood glucose level	mg/dl
A1c	Hemoglobin A1c (measure of blood glucose)	%
LDL	Low-density lipoprotein	mg/dl
HDL	High-density lipoprotein	mg/dl
Trig	Triglycerides	mg/dl
Weight	Weight	kg
Height	Height	m
BMI	Body mass index	kg/m^2
Creat	Serum creatinine	mg/dl
GlucMeds	Number of glucose meds patient is on	
BpMeds	Number of blood pressure meds patient is on	
LipidMeds	Number of statin meds patient is on	
GlucMoves	Number of glucose moves made on this patient	
BpMoves	Number of blood pressure moves made on this patient	
LipidMoves	Number of lipids moves made on this patient	
AdhMoves	Number of adherence moves made on this patient	
Prev*	Variables with a name beginning with Prev contain the corresponding value at the previous visit, e.g., PrevA1c holds the value A1c value from the previous visit. If this is the first visit, these variables contain the value at the current visit.	

An *action primitive* is a tuple (T, d) where T describes the condition to treat and d represents the number of days until the next visit. The possible values for T are listed in Table 31.

Table 31. Action (treatment) primitives

Treatment	Description
G	Treat blood glucose 3 oral medication and insulin
B	Treat blood pressure 6 oral medication classes
L	Treat lipids 2 oral medication classes
A	Treat adherence Refer to nurse educator or psychologist
N	Do nothing

The *state*, *action*, and *expectation function primitives* are described in Table 32.

Table 32. Function primitives

Function	Description
Primitive state functions (return a state)	
Cur(X)	Current value of a state variable (age, A1c, SBP, creatinine, etc.)
Prev(X)	Value of a state variable at time t-1; if t=0, return current value
Goal(X)	Goal value for a state variable
Primitive expectation functions (return a state)	
Exp(X, a)	Expected value based on an action a
Primitive action functions (return an action)	
PrevAct()	Last action (taken at time t-1); if t=0, return (N, 90)

Expectations return the expected state after an action was taken. For example, the mental model of a strategy using feedback control may contain an expectation that assumes the full effect of a glucose move is realized after 60 days, and that the effect is a decrease by 0.5%. Suppose a patient has a current A1c of 8.3%. For the mental model described here, Exp(A1c, (G,60)) evaluates to 7.8% but Exp(A1c, (G,45)) = 8.3%. The expectation represents only a single point on a dose response curve and, therefore, is represented by a step function (left side of Figure 52).

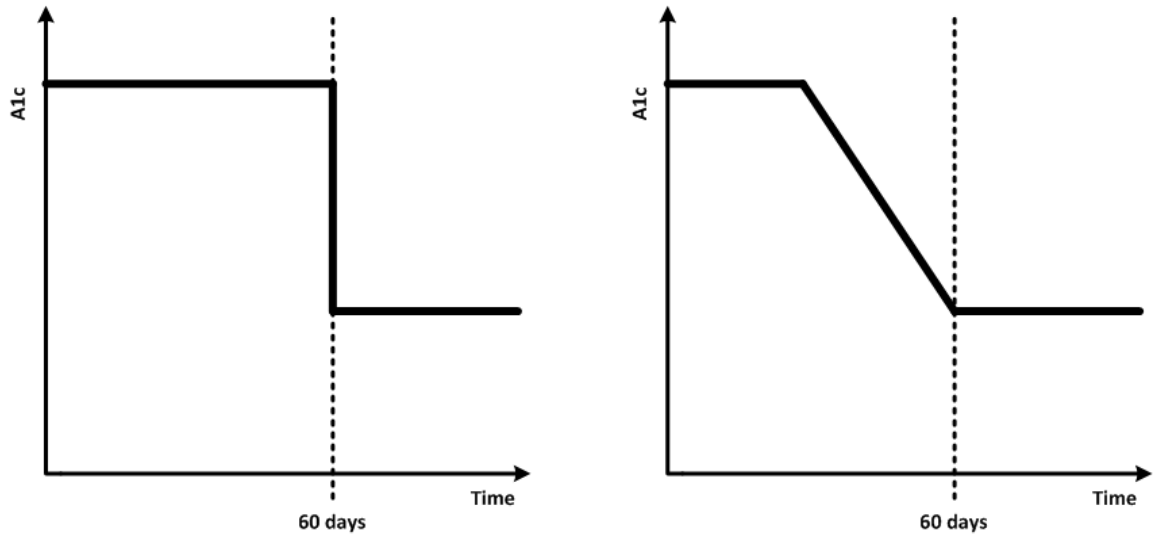

Figure 52. Expectations for A1c given a glucose treatment

In contrast, the mental model for a feedforward strategy may have the same 60-day expectation but its mental model contains dose responses for shorter time frames, so it might represent the time effects as piece-wise linear segments and estimate Exp(A1c, (G,45)) = 7.9% (illustrated in Figure 52 on the right).

Evaluation

To execute a strategy represented with the grammar described above, its predicates must be evaluated with respect to a given state. A vector containing the state is provided as input to the decision strategy. Expectations and goals are provided as functions available to the decision strategy, as discussed in Section 6.3. A challenge for evaluation is the specific representation of the state variables. For example, consider the predicate Cur(A1c) > Cur(SBP). A1c and SBP have different units of measure that cannot be compared and therefore the predicate could not be evaluated. Genetic programming

requires robust representations to allow arbitrary changes in programs due to reproduction and mutation. This is addressed in the case of state variables by representing them in terms of their distributions, which makes predicates such as the example above evaluable.

The value of a state variable is represented by its corresponding threshold in the *cumulative distribution function* (cdf). For example, a LDL value of 100mg/dl would be represented as the probability of observing LDL values lower than 100mg/dl in the distribution. This is illustrated in Figure 53 for a value x such that $P(LDL \leq x) = .3$.

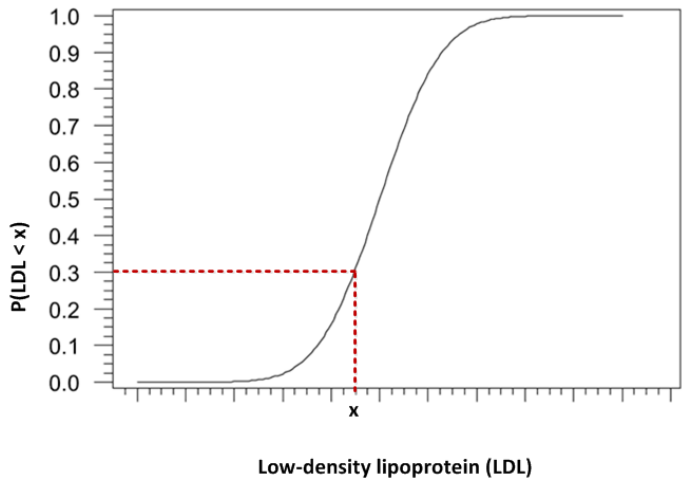

Figure 53. Example of representing an LDL by its corresponding threshold in the cumulative distribution function (cdf)

This representation can be applied to all state variables. Any arbitrary real value p between 0 and 1 can be translated to a threshold x for a given state variable. The result of an evaluated expression containing a state variable, for example, Cur(A1c) is returned in

this fashion. All random constants are chosen from the standard uniform distribution $U(0,1)$. Let R be such a random constant. If a predicate Cur(A1c) > R changes through mutation to Cur(SBP) > R, the predicate can still be evaluated.

In addition to the state variable representation, functions must be robust with respect to evaluation. For example, consider the function PrevAct() which returns the previous action taken. This function cannot return a value during the first visit because there is no previous action. Koza (1992) illustrates that it is desirable to use functions that are robust against such errors for genetic programming. If functions do have potential for errors, for example, division by zero, the function should be wrapped in one that can handle this situation, for example by defining a resulting value ε for a division by zero. Consistent with this approach, PrevAct(), for example, has been defined to return a default that constitutes a valid action (see Table 32).

A special case are *introns,* the equivalent to "junk DNA" that is inert in biology. Introns are actions that cannot be reached because their associated predicates always evaluates to true or false. Even though introns do not affect outcomes in a given strategy, the rules contained in the intron can become active again if they are incorporated into future strategies through recombination or mutation.

Control Parameters for Genetic Programming

Table 33 lists the control parameters for genetic programming, as described by Koza (1992), used in Study 3 to evolve decision strategies.

Table 33. Control parameters for genetic programming used in Study 3

Population size M = 100
Maximum number g of generations to be run = 101
Probability of crossover = 90%
Probability of reproduction = 10%
Probability of choosing internal points for crossover = 90%
Maximum size of S-expressions created during the run = stochastically determined, see Section 6.4.1
Maximum size for initial random S-expressions = stochastically determined, see Section 6.4.1
Probability of mutation (incl. permutation) = 10%
Frequency of editing = 0
Probability of encapsulation = 0
Condition for decimation = NIL
Decimation target percentage = 0
Generative method: described in Section 6.4.1
Basic selection method is tournament selection
Spousal selection method is tournament selection
Adjusted fitness is used
Over-selection is not used
Elitist strategy is not used

As mentioned above, the constraints on the space of strategies that can be evolved stem from the available primitives. The primitives provided here can describe strategies that use feedback as well as feedforward control. Feedback is incorporated by means of the Cur() function. Differential feedback can evolve by comparison of Cur() to Prev(). Feedforward strategies are represented by incorporating their predictions through expectation functions Exp().

An example of a strategy that cannot be represented given the constraints described above is the random choice strategy from Study 1. It cannot be represented given because the only random elements are constants. However, it would be possible to create this strategy by defining a new *action function primitive* RandAct() that returns a random action.