České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Lada O n d r á č k o v á

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Optimalizace inspekčních procesů klinických testů

Pokyny pro vypracování:

1. Seznamte se s problémem klinických testů.
2. Vytvořte model procesu klinických testů a popište jednotlivé fáze.
3. Formalizujte inspekční problém jedné fáze jako hru mezi farmaceutickou společností a doktory.
4. Navrhněte modely hry s různou složitostí (hra s nulovým vs. nenulovým součtem, současný tah vs. vůdce-následovník).
5. Implementujte algoritmy schopné vyřešit hry z bodu (4).
6. Změřte výkon algoritmů na syntetických datech.

Seznam odborné literatury:

[1] Shoham Y., Brown K.: Multiagent Systems. Cambridge University Press. 2009.
[2] U.S. National Institutes of Health. Clinical Trials. Online. Url: clinicaltrials.gov. 2014.

Vedoucí bakalářské práce: Ing. Ondřej Vaněk, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 14. 1. 2015

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Lada O n d r á č k o v á

Study programme: Open Informatics

Specialisation: Computer and Information Science

Title of Bachelor Project: Optimization of Clinical Trial Inspection Process

Guidelines:

1. Understand the problem of clinical trials.
2. Create a model of clinical trial process and describe each phase.
3. Formalize the inspection problem in one phase as a game between the pharmaceutical company and doctors.
4. Propose models of the game with differing complexity (zero-sum vs. non-zero sum, simultaneous move vs. leader-follower).
5. Implement algorithms able to solve the games defined in (4).
6. Evaluate performance of the algorithms on synthetic data.

Bibliography/Sources:

[1] Shoham Y., Brown K.: Multiagent Systems. Cambridge University Press. 2009.
[2] U.S. National Institutes of Health. Clinical Trials. Online. Url: clinicaltrials.gov. 2014.

Bachelor Project Supervisor: Ing. Ondřej Vaněk, Ph.D.

Valid until: the end of the summer semester of academic year 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

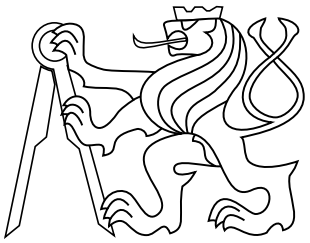
prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 14, 2015

bachelor's thesis

Optimization of Clinical Trial Inspection Process

Lada Ondráčková



May 2015

supervisor: Ing. Ondřej Vaněk, Ph.D.

Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....

Podpis autora práce

Acknowledgement

I would like to thank my supervisor, Ing. Ondřej Vaněk, PhD., for his patient guidance, willingness and assistance during the writing of my thesis. I would like to thank my family, for the support in my studies.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

Abstract

Pokud farmaceutická firma vyvíjí novou látku, pak je pro ni velice důležité, aby obdržela korektní data z klinického testu od lékaře, který kontrolovalje v průběhu testu vliv látky na pacienty. Tedy, pokud si chce být farmaceutická firma jista, že od lékaře obdrží data, která lékař nezměnil, pak ho musí průběžně kontrolovat.

V této práci jsme se zaměřili na optimální plánování inspekcí v jedné fázi klinických testů. Formalizovali jsme tento problém jako hru mezi farmaceutickou firmou a lékařem, kde inspektoři farmaceutické firmy kontrolují lékaře tak, aby maximalizovali pravděpodobnost, že lékaři posílají pouze korektní data. Nejprve jsme formalizovali problém jako časově nezavilý, od kterého jsme odvodili časově závislý model pro plánování inspekcí. V časově závislém modelu bereme v úvahu, že každý týden ovlivňuje rozhodnutí o účinnosti látky s jinou váhou. Optimální plán inspekcí hledáme v podobě Nashova a silného Stackelbergova equilibria podle struktury užitkové funkce hráčů.

Nad rámec zadání jsme dekomponovali problém rozdělování celkového rozpočtu na inspekce, kde plánujeme pokrytí inspekcemi celého klinického testu přes všechny fáze.

Nakonec jsme zhodnotili naimplementované algoritmy na daných scénářích a ukázali vlastnosti řešení a škálovatelnosti algoritmu hledajiciho řešení.

Klíčová slova

Klinické testy; teorie her; inspekční procesy; optimalizace

Abstract

If the pharmaceutical company develops new drug, then it is very important to observe correct data from the clinical trial from the doctor, who controls drug's effect on the participants in the trail. If the pharmaceutical company wants to be certain that the doctor reports correct data then they have to inspect him.

In this thesis, we focus on optimal scheduling inspections in one phase of the clinical trial. We formalize problem as the game between the pharmaceutical company and the doctor, where the pharmaceutical company though the inspector wants to protect data from the testing from the doctor's changes. Firstly, we formalize the game as time independent model and then we extend time dependent model for scheduling inspections. Time dependent model respects that every week in trial has different weight for decision about the efficiency of the tested drug. The optimal schedule for inspections is found using Nash and Stackelberg equilibrium with dependence on the utility function of the agents.

Out of the assignment, we decompose the problem of budget division, where we plan coverage of all clinical trial by the inspections.

Finally, we evaluate implanted algorithms on defined scenarios and show performance and scalability of the algorithm.

Keywords

Clinical trials; game theory; inspection processes; optimization

Contents

Abbreviations

SSE	Strong Stackelberg Equilibrium
NE	Nash Equilibrium
LP	Linear Programming
FDA	The Food and Drug Administration

1 Intro

Pharmaceutical industry is area which is based on developing new drugs and selling them. In this thesis, we focus on the clinical trials. The clinical trial is part of testing new drug, where the drug for testing is already developed and tested in preclinical testing. The clinical trial is divided into Phases. Each phase tests the drug specifically for efficacy, dosage and side-effects. Volunteers participate in testing each Phase and they test the drug. They take the drug for Time period and they are controlled by the doctor who controls them in control weeks. The doctor controls how the drug affects their state of health.

In this thesis, we focused on the detection of fraud in one Phase of clinical trial. Fraud can be conducted by the participants or by the doctor.

The doctor controls participants and prevents participant's fraud. He reports about the state of health of the participants and this reports are very important for the pharmaceutical company. But even doctor can perform fraud due to competitive or adversarial reasons. And because he collects the data about the group of participants his changing reports can significantly affects the decision, which the pharmaceutical company makes about the efficacy and the future of the drug.

We focused on the doctor's fraud, which has a significant importance for the pharmaceutical company. The doctor's fraud can be prevented by the inspections. The inspector inspects doctor in control weeks, when the doctor controls the participants of the testing. The inspector wants to detect if the doctor performs fraud and he wants to prevent doctor's fraud by inspecting him frequently. Nowadays, if the pharmaceutical company wants to know that they observe correct data from the doctor then inspections should be every control week in the Time period of the Phase.

The goal of this thesis was to plan the optimal schedule of inspections for the inspector for one Phase of clinical trial if the pharmaceutical company does not want to inspect every control week in the Time period or does not have the budget to inspect the doctor every control week in the Time period. We expected that if the doctor is inspected and performs fraud then the inspector detects doctor's fraud.

We formalized the inspection problem of one Phase of clinical trial as a security game, where the inspector is defender who wants to detect and prevent doctor's fraud. The doctor is attacker who wants to perform fraud by changing the results of the Phase of clinical trial. Then we searched for the solution of the security game using Nash or Strong Stackelberg equilibrium in dependency on structure of utility function of agents.

In our work, we created two utility models of security games that optimally solved the inspection scheduling problem. Firstly, we created time independent model. In this model, the type of Phase is defined by the number of control weeks in the Time period of the Phase and limited budget is represented by maximal number of inspections in the Phase.

Secondly, we extended this time independent model to time dependent model. The time dependent model reflects that the control weeks in the Time period of Phase, when participants are controlled by the doctor, have not the same weight for decision about efficacy of the drug. For example, some control weeks are more focused to control patients if they use the drug in compliance with using the drug and some are important

to check the drug effect on the state of health of participants. Time dependent model uses the same specification of Phase and representation of limited budget as time independent model plus the time dependent model specifies the importance of every control week in the Time period of the Phase of clinical trial. Both models expect that the doctor is motivated to perform fraud.

We evaluated the proposed algorithms on synthetic data on which we measured the influence of single parameters on scalability and quality of the obtained solutions. Concretely, we evaluated our game theoretical solution and other simplification strategies on the same scenario and then we compared the solutions. We find out that the the solution computed by our model gave us the best result in comparison with other simplification strategies. Then, we wanted to find the instances of the inspection problem which are the hardest to solve, hence we computed the $(d : s)$ ratio.

Out of the bachelor project assignment, we created model of optimal budget division. This model reflects problem how to divide optimally budget for inspection into Phases of clinical trial if the pharmaceutical company knows with high probability that the drug is effective and we expected that somebody wants to thwart clinical trial of the drug. In this model, we used previously defined model for planning inspections in one Phase of clinical trial.

1.1 Goals of the thesis

This thesis contains goals which are described in following description.

Understand the problem of clinical trials

The problem of fraud in clinical trials is decomposed in Chapter (4). We describe that clinical trial is liable to frauds, which can has various types. Techniques for prevention of fraud and methods how the control organization as FDA detect fraud in clinical trials.

Create a model of clinical trial process and describe each Phase

Process of clinical trial is described in Chapter (4) where the process is decomposed and every Phase is briefly introduced. Then the model of clinical trial process is used for model of budget division in Chapter (5).

Formalize the inspection problem in one Phase as a game between the pharmaceutical company and doctors

The game models used in this thesis are formalized in Chapter (5). We formalize the inspection problem in one Phase as a game between the pharmaceutical company and doctors in two utility models. The first is time independent model and the second is time dependent model.

Propose models of the game with differing complexity (zero-sum vs. non-zero sum, simultaneous move vs. leader-follower)

Specification of different game models with different complexity is compared in Chapter (5). Firstly, we compare zero-sum games and their types and then non-zero sum games.

Implement algorithms able to solve the games defined in (4)

We solve the model of inspection problem in one Phase as Stackelberg Security game. The main part of the implementation is described in Chapter (6).

Evaluate performance of the algorithms on synthetic data

Finally, we evaluated implement algorithms and this evaluation is shown in Chapter (7).

2 Methods and techniques

The main part of technical work presented in this thesis is based on the Game theory, linear programming and optimization. This chapter introduces theoretical background used for solving and modeling the inspection scheduling problem one Phase of clinical trials.

2.1 Game theory intro

Game theory is a mathematical framework for capturing interaction among independent, self-interested agents [1]. *Self-interested agent* means that each agent has his own description of which states of the world he prefers. His acts can include good things or bad things to other agents. Agent acts in an attempt to bring about these states of the world.

Agent should be able to appraise all states of the world. Value of the state is represented by a *utility function*. A utility function is a mapping from states of the world to real numbers. These numbers are interpreted as measures of an agent's level of satisfaction in the given states. When the agent is uncertain about which state of the world he faces, his utility is defined as the expected value of his utility function with respect to the appropriate probability distribution over states.

Agents are parts of the *game*. The game is interaction between agents in defined area of model world.

Every agent in a game is a self-interested and rational agent who would like to maximize his utility. It means he would like to execute *actions* that maximize utility for him. The set of actions of the player is set of all possible state transitions that agent can play.

2.2 The normal form game

In the normal form game, every agent has utility functions and wants to maximize expected value of his utility function [1]. He chooses a single action that maximizes expected utility. This suggests that acting optimally in an uncertain environment is conceptually straightforward at least as long as the outcomes and their probabilities are known to the agent and can be succinctly represented. However, situation is more complicated when the world contains two or more utility-maximizing agents whose actions can affect each other's utilities.

The normal form representation is also known as the strategic form and is arguably the most fundamental in game theory. The normal form game does not contain any kind of uncertainty. Every player has to have representation of utility function for every state of the world, in the special case where states of the world depend only on the player's combined actions in game written in this way.

2.2.1 Definition of Normal from game

Definition 1. *A (finite,n-person)normal-form game is a tuple (N ,A ,u), where [1]:*

- *N is a finite set of n players, indexed by i ;*
- $A = A_1 \times \cdots \times A_n$ *where* A_i *is a finite set of actions available to player i. Each vector* $a = (a_1, ..., a_n) \in A$ *is called an action profile;*
- $u = (u_1, ..., u_n)$ *where* $u_i : A \longmapsto \mathbb{R}$ *is a real-valued utility (or payoff) function for player i.*

A natural way to represent games is via an n-dimensional matrix. The two-dimensional matrix is used for two-player game in this thesis. Each row corresponds to a possible actions (strategies) for player 1, each column corresponds to a possible action (strategies) for player 2, and each cell corresponds to one possible outcome. Each player's utility for an outcome is written in the cell corresponding to that outcome in player's two-dimensional matrix.

2.2.2 Types of strategies

One type of strategy for agent is to select a single action and play it. This type of strategy is called a pure strategy, and the notation is the same as for actions to represent it. A choice of pure strategy for each agent is called a pure-strategy profile.

Players could also follow another and use less obvious type of strategy like randomizing over the set of available actions according to some probability distribution. Such a strategy is called a mixed strategy. We define a mixed strategy for a normal-form game as follows.

Definition 2. *Mixed strategy: Let (N,A,u) be a normal form game, and for any set X let* $\Pi(X)$ *be the set of all probability distribution over X. Then the set of mixed strategies for player i is* $S_i = \Pi(A_i)$.

Definition 3. *Mixed strategy profile: The set of mixed-strategy profiles is simply the Cartesian product of the individual mixed-strategy sets,* $S_1 \times \cdots \times S_n$.

2.3 Nash equilibrium

Nash equilibrium is the most influential solution concept in game theory.

The most important is that if an agent knows how the others are going to play, his strategic problem would become simple. The problem would be simplified to the single-agent problem of choosing a utility-maximizing action. Formally, define $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ...s_n)$, a strategy profile s without agent i's strategy. We can write $s = (s_i, s_{-i})$. If the agents other than i (whom we denote $-i$) were to commit to play s_{-i}, a utility-maximizing agent i would face the problem of determining his best response.

Definition 4. *Best response: Player i's best response to the strategy profile* s_{-i} *is a mixed strategy* $s_i^* \in S_i$ *such that* $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s{-}i)$ *for all strategies* $s_i \in S_i$.

The best response may not be unique. The best response is unique only in extreme case that it is a pure strategy. The number of best responses is always infinite in other cases. When the support of a best response s^* includes two or more actions, the agent must be indifferent among them — otherwise, the agent would prefer to reduce the probability of playing at least one of the actions to zero. It means, any mixture of these actions must also be a best response, not only the particular mixture in s^*. Similarly, if there are two pure strategies that are individually best responses, any mixture of the two is necessarily also a best response. In general an agent will not know what

strategies the other players plan to play. The result is, the notion of the best response is not a solution concept, because it does not identify an interesting set of outcomes in this general case. But we can use the idea of best response to define what is arguably the most central notionin noncooperative game theory, the Nash equilibrium.

Definition 5. *Nash equilibrium: A strategy profile $s = (s_1, ..., s_n)$ is a Nash equilibrium if, for all agents i, s_i is a best response to s_{-i}.*

Intuitively, a Nash equilibrium is a stable strategy profile. Thus no agent would want to change his strategy if he knew what strategies the other agents were following.

Theorem 1. *Theorem (Nash, 1951) : Every game with a finite number of agents and action profiles has at least one Nash equilibrium.*

Proof. Proof is described in [1]. □

2.4 Strong Stackelberg equilibrium

In Multiagent systems, strategic settings are often analyzed that the agents choose their strategies simultaneously. However, strategies could not be always selected in such a simultaneous manner. In many real-world settings oftentimes, one agent is able to commit to a strategy before the other agent makes a decision [2]. The agent, who is able to commit to a strategy before the other agent is called the leader and the other agent, is called the follower. In a Stackelberg model, the leader chooses its strategy first, and the follower chooses a strategy after observing the leader's choice. This can happen due to variety of reasons. Commitment power has a profound impact on how the game should be played. In general, if commitment to mixed strategies is possible, then it never hurts, and often helps, to commit to strategy [3].

Theorem 2. *In 2-agent normal-form games, an optimal mixed strategy to commit to can be found in polynomial time using linear programming.*

Proof. For every pure follower strategy t is computed a mixed strategy for the leader such that [2]. Playing t is a best response for the follower, and under this constraint, the mixed strategy maximizes the leader's utility. This mixed strategy can be computed using the following linear program (1). Where S is set of leader's pure strategies and u_l is his utility function. T is set of follower's pure strategies and u_f is his utility function.

$$\begin{aligned} \forall t \in T \text{ maximize} \quad & \sum_{s \in S} p_s u_l(s,t) \\ \text{subject to} & \\ \forall t' \in T \quad & \sum_{s \in S} p_s u_f(s,t) \geq \sum_{s \in S} p_s u_f(s,t') \\ & \sum_{s \in S} p_s = 1 \end{aligned} \tag{1}$$

This program may be in-feasible for some follower strategies t. For example, if t is a strictly dominated strategy. However, the program must be feasible for at least some follower strategies. From these follower strategies, it can be chosen a strategy t^* that maximizes the linear program's solution value.

Then, if the leader chooses mixed strategy which corresponding the optimal settings of the variables p_s for the linear program for t^* and if the follower plays t^*, then it constitutes an optimal strategy profile. □

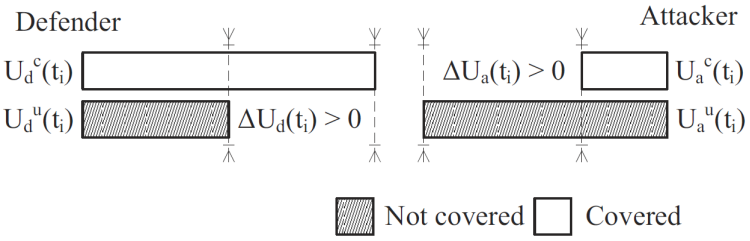


Figure 1 Utility function structure of Security games [4]

2.4.1 Security games

In security games there are set of possible targets $T = \{t_1, t_2, \cdots, t_n\}$ and two agents — defender and attacker. In contrast with other types of games, all the security game models have specific utility function as shown in Figure (1). Where $U_d^c(t_i)$ represents defender's utility if t_i is attacked by the attacker while t_i is covered by some defender resource and $U_d^u(t_i)$ if t_i is not covered by any defender's resource. Attacker's utility $U_a^c(t_i)$ represents if t_i is attacked by the attacker while t_i is covered by some defender resource and $U_u^a(t_i)$ if t_i is not covered by any defender's resource. Difference between defender's covered and uncovered utilities is represents as $\Delta U_d(t_i) = U_d^c(t_i) - U_d^u(t_i)$. Similarly, the difference for the attacker is $\Delta U_a(t_i) = U_a^u(t_i) - U_a^c(t_i)$. As a key property of security games, it is assumed $U_d(t_i) > 0$ and $U_a(t_i) > 0$ [4].

Stackelberg Security games

In Stackelberg security games defender is the leader and attacker is the follower of the game. Defender wants to protect these set of targets and attacker wants to attack these targets. Each of these targets has a unique profit and loss to both. To protect these targets, the defender has a set of strategies. He counts the best one via SSE which maximizes his reward and he commit to these strategy. He expects that attacker (follower) observe his strategy and then attacker choose targets to attack. Even attacker wants to maximize his profit. Thus, if the attacker attacks target which is not protected by the defender, then attacker has a profit from these action and defender loose this action else vice verso.

2.4.2 Using Strong Stackelberg equilibrium in this thesis

Lets imagine the following situation. The pharmaceutical company has doctors, who control patients in clinical trials. The pharmaceutical company conducts a lot of clinical trials and the company cooperates with the same doctors repeatedly. Sometimes, for reasons discussed further, the doctors are inclined to perform frauds. Thus doctors are inspected repeatedly by the inspector and they can learn the inspection strategy through observation. In the modeling game for planning inspections, we have to expect that the doctor knows inspector's strategy.

Finally in this thesis, we use the Stackelberg Security game model. The inspector acts first as a leader by committing to an inspection strategy and the doctor as follower chooses when to cheat after observing the inspector's choice. Targets are represented as weeks when the doctor can cheat. The typical solution concept applied to these games is Strong Stackelberg Equilibrium, which assumes that the inspector will choose an optimal mixed strategy based on the assumption that the doctor will observe this strategy and choose an optimal response.

2.5 Linear programming

Linear programming (LP) is an essential optimization technique. LP solves the problem represented by model. Model is structure of problem which has been built with purpose exhibiting features and characteristic of the problem. More about model building is described in [5].

The concept of a Linear program contains following components. *Decision variables* are quantities to be determined. How the decision variables affect the cost or value to be optimized is represented via an *Objective function*. The objective function is a linear function which can be minimized or maximized and which is limited by *Constrains*. Constrains are affine functions which represent linear relationships and influence how the decision variables use resources, which are available in limited quantities.

If we solve some model, then solving a linear program has three possible types of solutions [6]. Firstly, model has at least one optimal solution. Secondly, model has empty set of feasible solutions. It indicates that constrains are in contradiction. Thirdly, model is unbounded and model's objective function with given constrains can be unlimitedly improved.

A Linear program can be represented by several forms [7]. The first form is to represent model in a *general form* (2). The general form representation permits the objective function to be maximized or minimized, allows both inequalities and equality constrains and puts no constraints on the values of the variables other than the constraints that appear in the program. Another possibility is a *canonical form* (3) which has following regulations. Constrains are allowed only in $Ax \leq b$ from, the objective function has to be maximized and it is required that decision variables are non-negative.

$$\begin{array}{rrcl} \text{max or min} & c^t x & & \\ \text{subject to} & Ax & \geq & b \\ & Ax & \leq & b \\ & Ax & = & b \\ & x & \in & \mathbb{R} \end{array} \tag{2}$$

$$\begin{array}{rrcl} \text{max} & c^t x & & \\ \text{subject to} & Ax & \leq & b \\ & x & \geq & 0 \\ & x & \in & \mathbb{R} \end{array} \tag{3}$$

2.6 Decision tree

Decision trees are great tool for assistance in choosing between several options of an action [8]. They can show us a balanced picture of the risks and rewards associated with each possible course of action and allow us to analyze the possible consequences of a decision fully. It helps us to make the best decisions on the basis of existing information and best guesses.

Firstly, it is necessary to draw the decision diagram of some problem [9]. Decision diagram starts in the root node and branch out via actions to another nodes and finally to leaves as is shown in Figure 9. Node represents result of an action and it can be a decision node or an uncertainty node. In decision node, we have to decide which action will follow. And in uncertainty node, we do not know which action will follow but every outgoing action from uncertainty node has a probability that occurs. If the probability is in percentages, probabilities of all actions from one node have to sum up to 100%.

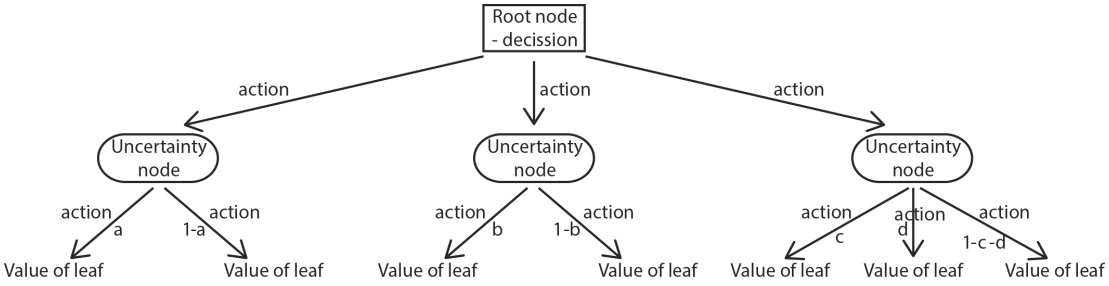


Figure 2 Decision tree

Now the decision diagram is ready for the evaluation. If the diagram contains some uncertainty nodes, we have to evaluate or estimate the probability of each outgoing action. Then the leaves node should be assigned with some value, the best leafs node with the highest value and the worst leaf outcome with the lowest value. If we have done previous steps, we can start calculating the values that will help us make our decision. We evaluated leaf nodes and we will calculate the value of nodes back towards the root node.

The value of an uncertainty node is calculated by summing up all values of outgoing actions which are multiplied by their probability.

When we evaluate the decision node, we can easily choose the max value of outgoing action. But actions have oftentimes some cost. Thus we have to subtract the cost of action from every value of outgoing node and then find the maximal value of outgoing node. Then maximal value of outgoing action is value of the node.

At the end we have evaluated outgoing nodes from the root node. So we can see the reward of each outgoing action and decide for the best action.

2.7 CPLEX

Linear programming was revolutionized when CPLEX software was created. The CPLEX was developed by Robert E. Bixby in 1987, it was distributed by ILOG company and now the CPLEX is distributed by IBM since 2009. The CPLEX is high performance solver for Linear Programming (LP), Mixed Integer Programming (MIP) and Quadratic Programming (QP/QCP/MIQP/MIQCP) problems.

For problems with linear constraints, CPLEX uses a simplex method or a primal-dual interior point method to solve the problem. The CPLEX package contain following four distinct methods for solving problem[10]. First, a primal simplex algorithm that first finds a solution feasible in the constraints, then iterates toward optimality. Second, A dual simplex algorithm that first finds a solution satisfying the optimality conditions, then iterates toward feasibility. Third, a network primal simplex algorithm that uses logic and data structures tailored to the class of pure network linear programs. Fourth, a primal-dual interior-point algorithm that simultaneously iterates toward feasibility and optimality, optionally followed by a primal or dual crossover routine that produces a basic optimal solution.

The simplex algorithm is fundamental part of CPLEX, which was named for the simplex method as implemented in the C porigrameng language. The simplex algorithm is more described by M. Trick [11].

The primal-dual interior point algorithm for linear programming used in CPLEX was introduced by Megiddo. He use logarithmic barrier methods to solve the primal and dual problems simultaneously. He describes this algorithm in book [12].

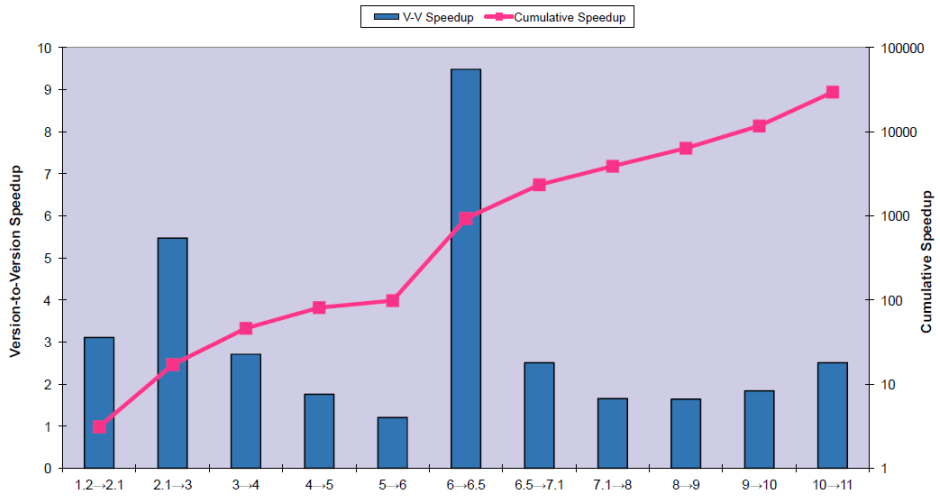


Figure 3 CPLEX Performance [13]

E. Bixby recompiled [13] each of the corresponding twelve CPLEX released versions from Version 1.2 through CPLEX 11 and he compared them. He shows improvement of scalability during the time which is shown in Figure 3. The scale on the left refers to the bars in the bar chart and shows version to version speed up. The scale on the right to the piecewise-linear line through the middle and shows cumulative speed up. We can see that bar comparing CPLEX 3.0 to 2.1 stands out, because it has Version-to-Version Speedup of nearly 5.5. It corresponds to the maturity of the dual simplex algorithm. Second and the biggest stand out bar compares CPLEX 6.5 to 6.0. These big speedup was caused by ability to solve real-world MIPs. The piecewise-linear line through the middle shows us that overall speedup factor from CPLEX 1.2 to CLEX 11 is almost 100,000.

3 Related work

3.1 Stackelberk games

There has been significant recent research interest in game-theoretic approaches to security at airports, ports, transportation, shipping and other infrastructures. Many of these problems have used Stackelberg game framework to model interactions between defenders and attackers.

Stackelberg games are at the heart of decision-support applications like ARMOR, IRIS, GUARDS and PAWS.

3.1.1 ARMOR

Pita at al. [14] proposed protecting national infrastructure such as airports including tasks such as monitoring all entrances or inbound roads and checking inbound traffic. Where limited resources imply that it is typically impossible to provide full security coverage at all times. In particular, they proposed a software assistant agent called ARMOR (Assistant for Randomized Monitoring over Routes). They mapped the problem of security scheduling as a Bayesian Stackelberg game and they solved it via an algorithm called DOBSS (Decomposed Optimal Bayesian Stackelberg Solver). In summary, they modeled problem via Stackelberg game with two types of agents. The police force is a leader and their adversaries are followers. They assume that there are m different types of adversaries, each with different attack capabilities, planning constraints, and financial ability. Each adversary type observes the police force checkpoint policy and then decides where to attack. Attacker's targets are inbound roads 1 through n. The police force has picked up p resources-roads to place checkpoints. Thus, their strategy is all combinations of p checkpoints. Each adversary type can choose strategy and decide to attack one of the n roads or possibly not attack at all. If the police force selects road i to place a checkpoint on and adversary type l selects road j to attack then both receive differed rewards. These rewards depend on three considerations: (i) the chance that the Los Angeles World Airport police checkpoint will catch the adversary on a particular inbound road; (ii) the damage the adversary will cause if it attacks via a particular inbound road; (iii) type of adversary, i.e. adversary capability. For example, if Police force catches the adversary then it is positive reward for police and negative reward for adversary.

ARMOR has been successfully deployed since August 2007 at the Los Angeles International Airport (LAX) to randomize checkpoints on the roadways entering the airport and canine patrol routes within the airport terminals.

3.1.2 IRIS

Tsai at al. [15] proposed protection of transportation networks such as airplanes which carry millions of passengers per day. It makes them a prime target for terrorists and extremely difficult to protect for law enforcement agencies. They implement IRIS (Intelligent Randomization In Scheduling) system based on strategic randomization. IRIS

is modeled as a Stackelberg game, with law enforcement agencies as leaders that commit to a flight coverage schedule and terrorists as followers that attempt to attack a flight. For solving this class of Stackelberg game was used ERASER-C algorithm. IRIS is a scheduling assistant for the Federal Air Marshals (FAMS) which provides a game-theoretic solutions similar in spirit to the ARMOR.

3.1.3 GUARDS

Pita at al. [16] developed a new application called GUARDS (Game-theoretic Unpredictable and Randomly Deployed Security) to assist in resource allocation tasks for airport protection at over 400 United States airports. TSA is charged with protecting over 400 airports in the US. The key challenge is how to intelligently and predictably deploy limited security resources. They lead to a new game model called "Security Circumvention Games" (SCGs) and they work with Stackelberg game with two agents. The leader of the game and the defenders of the airports is United States Transportation Security Administration (TSA). The follower of the game is TSA's potential adversary. Defender has set of pure strategies and he is able to execute variety of security activities in the set of different areas. Follower has set of pure strategies where each of them corresponds to selection of a single area and a specific mode of attack. They use DOBSS Stackelberg game solver.

3.1.4 PAWS

Ford at al. [17] formulates the wildlife crime problem. The Protection Assistant for Wildlife Security (PAWS) generates optimized defender strategies for use by park rangers. PAWS implements a novel adaptive algorithm that processes crime event data, builds multiple human behavior models, and, based on those models, predicts where adversaries will attack next. These predictions are then used to generate a patrol strategy for the rangers that can be viewed on a GPS unit. They model security game as a Bayesian Stackelberg game with infinite types, where the leaders are the rangers and the followers are the poachers.

4 Clinical trials intro

In this chapter, we want to introduce the background of clinical trials. The area of clinical trials is primary created by pharmaceutical companies and by the control's regulations of clinical trials. Firstly, we introduce the process of clinical trials and then we focus on introduction to fraud and misconduct in clinical trials and how it can be detected and prevented. All these general information help us to imagine process of developing new drug in pharmaceutical area.

4.1 Review intro

Process of discovering and developing new drug is long, complex and expensive as is shown in Figure 4.

Firstly, new drug is developed be the research community [18].

Thousands of drugs are discovered in research but only a few hundred drugs are suitable to continue into preclinical testing. In preclinical testing, drug are tested on animals or in laboratories. Tests should determinate whether a drug is suitable for human testing.

If the drug successfully finish preclinical testing then the drug can continue to clinical trial. In clinical trials, the drug is tested on human volunteers—participants. The process of clinical trial takes approximately six to seven years. The process of clinical trial is divided into several phases. The drug must successfully complete all of these phases and then it can be submitted to the FDA for review.

If the drug successfully completed first three phases of the clinical trial then it indicates that the drug is safe and effective. Then the pharmaceutical company can submit a New Drug Application to the FDA. The pharmaceutical company has to make available for FDA the data from the whole process of previous testing. Scientists at the FDA review all the results from previous testing and then they decide whether to grant approval that the drug is safe and it has declared effect.

4.2 The specific explanation of clinical trials

A clinical trial represents an international trial involving human subjects, who participate Phases of clinical trials [19]. A clinical trial does not include the use of drug in the normal course of medical practice or non-clinical laboratory study. Clinical trials are tests of vaccines, drugs, or new uses for existing drugs. Tests should detect efficiency, safety, side-effects and another specification of the drug.

4.2.1 Definition of terms

Vaccine is a biological preparation of weakened or killed forms of the microbe, its toxins or one of its surface proteins by Britannica [20]. Vaccination is process administering vaccine by injection or orally. The main importance of vaccine is primarily to prevent disease. Vaccine must be effective and harmless.

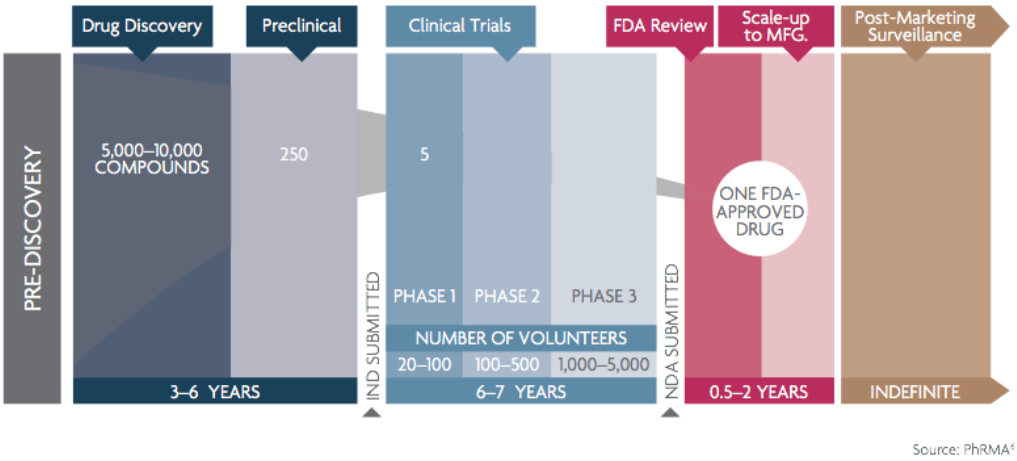


Figure 4 Drug approval process [18]

Drug is by FDA [21] a substance recognized by an official pharmacopoeia or formulary which is intended for use in the diagnosis, cure, mitigation, treatment, or prevention of disease. It is a substance other than food intended to affect the structure or any function of the body or a substance intended for use as a component of a medicine but not a device or a component, part or accessory of a device.

FDA (The Food and Drug Administration) is a federal agency of the United States Department of Health and Human Services. The FDA is responsible for protecting and promoting public health through the regulation and supervision of pharmaceutical drugs and others.

4.3 Clinical trial

Clinical trial is divided into Phases. Every Phase tests different criteria on groups of volunteers. Number of volunteers in groups depends on specification of each Phase. Volunteers are generally paid for participating in the testing.

4.3.1 Phase I (Checking for safety)

Phase I assesses the safety of a drug. Phase I is an initial phase of testing. Phase I can take from six to twelve months or more to complete. The drug is tested on a group of 20-100 participants [22].Participants in this Phase are mainly healthy. The study is designed to determinate medicine safety, reaction of the body to medicine, indication of medicine, expected effects and side effects of the drug. About 70% of experimented drugs pass this phase of testing successfully [22].

4.3.2 Phase II (Checking for efficacy)

Phase II studies the efficacy of a drug. The Phase normally takes from six to ten moths or more and involves group of 100-500 participants [22]. Participants for testing vaccines are mainly healthy. Participants for the drug testing have the disease or condition the medicine is designed to treat. The Phase is designed to detect the drug's effects, safety of the drug, side effects and indication of the drug. About 33% of experimented drugs successfully complete both, Phase I and Phase II.

4.3.3 Phase III

Phase III is the most expensive part of clinical trial. Testing can take from 1 to 4 years. Medicament is tested on a group of 1000-5000 participants [22]. Participants for testing medicine have the disease or condition the drug is designed to treat. Participants for testing vaccine are mainly healthy or have the disease or condition the drug is designed to treat. The study is designed to control how the drug's effects are good, safety of the drug, side effects and indication of the drug. The main difference between Phase II and Phase III is the number of participants and total complexity. From the drugs that enter Phase III then from 70% to 90% drugs successfully complete this phase of testing [22]. The pharmaceutical company can request FDA approval for marketing the drug after the drug pass Phase III. It means, the pharmaceutical company submits a New Drug Application.

4.3.4 Phase IV

Phase IV is conducted after a drug has been approved for consumer sale. Pharmaceutical companies continue in research to get more information about the drug or the vaccine and its safety, side effects and effectiveness [22]. Pharmaceutical companies can also compere medicament with other medicaments already in the market.

Marketed products are also studied for new indications. Thousands of people usually participate in ongoing trials.

In this thesis, we does not use the Phase IV for future decomposition of the process of clinical trial. We focus on the part of clinical trial before the drug is approved by the FDA.

4.4 Drug approval process costs

Facts about budget are explained by Roy [23]. The budget invested into the pharmaceutical industry has quickly increased in last 40 years. The equivalent of $100 million in today's dollars was spent for research and development of the average drug approved by FDA in 1975. The budget of $300 million was spent in 1987 and $1.3 billion in 2005. The budget is definitely larger today as can be seen in Figure 5.

Matthew Herper found that 12 leading Pharmaceutical companies had spent $802 billion to gain approval for just 139 drugs from 1997 to 2011 [23]. It means, a staggering $5.8 billion per drug.

The budget increased due to the regulations of testing new drugs on human volunteers in Phase III of clinical trial. Phase III has become larger and more complex.

The Tuft's group has shown, that the average length of a clinical trial increased by 70%, the average number of routine procedures per trial increased by 65% and the average clinical trial staff work burden increased by 67% in research from 1999 to 2005. The increasing trend is shown in Figure 6.

Criteria for selection participants in clinical trial has been considerably tightened and the number of volunteers admitted into trials declined by 21%. More than 30% of participants drop out clinical trial before completion.

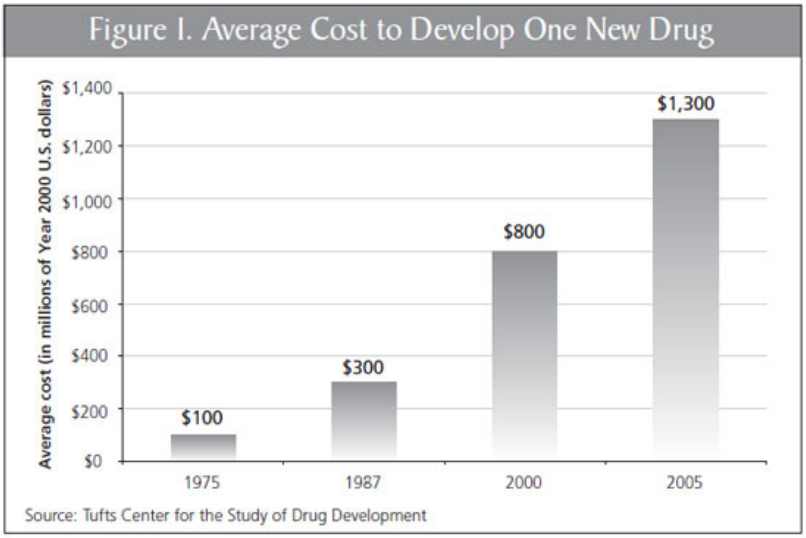


Figure 5 Average Cost to Develop One New Drug [23]

Table I. Changes in Clinical Trials: Resources, Length, and Participation

Function	1999	2005	Percent Change
Median procedures per trial protocol (e.g., blood work, routine exams, x-rays, etc.)	96	158	65%
Average clinical trial staff work burden, work-effort units	21	35	67%
Average length of clinical trial, days	460	780	70%
Clinical trial participant enrollment rate (% of volunteers meeting trial criteria)	75%	59%	-21%
Clinical trial participant retention rate (% of participants completing trial)	69%	48%	-30%

Source: Tufts Center for the Study of Drug Development, Impact Report 10, No. 1 (2008)

Figure 6 Changes in Clinical Trials [23]

Pharmaceutical companies in research and development spend 40% of expenditures to Phase III of clinical trials. Overall expenditures include hundreds of pharmaceutical candidates that never reach Phase III tests. Phase III clinical trials represent 90% or more of the cost of developing an individual drug. Expenditures are written out in Figure 7.

4.5 Fraud and misconduct in clinical trials

Fraud and misconduct in clinical trials are widespread problem. Good clinical practice is used international guideline for conduct of clinical trials. But internationally harmonized framework for managing research fraud and misconduct is unavailable. It makes clinical research vulnerable area to commit fraud [24].

Table 2. R&D by Function, PhRMA Member Companies, 2009

Function	Dollars (xMM)	Share of Total	Probability of FDA Approval
Prehuman/preclinical	$11,717.4	28.6%	8%
Phase I	$ 3,752.9	9.2%	21%
Phase II	$ 7,123.7	17.4%	28%
Phase III	**$16,300.1**	**39.8%**	**58%**
Approval	$ 2,046.9	5.0%	90%
Total R&D up to FDA approval	**$40,941.0**	**100.0%**	
Phase IV	$ 5,302.7	13.0%	
Uncategorized	$ 197.8	0.5%	

Source: PhRMA Annual Member Survey, 2011; DiMasi et al., J Health Econ 22(2003):151–85

Figure 7 Expenditure [23]

4.5.1 Intro

Several studies have found that more than 40% of researchers were aware of misconduct but did not report it. Sheehan et al. reported in 2005 that 17% of surveyed authors of clinical drug trials reported that they personally knew of fabrication in research occurring over the previous 10 years [24]. Clinical trials are controlled by audits and inspections. It should prevent fraud and misconduct.

Fraud and misconduct can lead to study losing its credibility, to ineffective or harmful treatment being available or patients being denied of effective treatment.

4.5.2 Definition of terms

Fraud and misconduct are two terminologies often used interchangeably. Both is a violation of the standard codes of scholarly conduct and ethical behavior in scientific research. But there is difference between these terms.

Misconduct may not be an intentional action, rather an act of poor management. It also includes failure to follow established protocols if this failure results in unreasonable risk or harm to humans [24].

Fraud should have an element of deliberate action, which is not the case with misconduct. Definition of the fraud is defined in court as "the knowing breach of the standard of good faith and fair dealing as understood in the community, involving deception or breach of trust, for money." [24].

4.5.3 Types of fraud

Fraud can be fabrication, falsification, and plagiarism of data or even deception in conduct by Gupta [24]. Fabricating data is creating a new record of data or results. Informed consent Forms and Patient diaries are the most commonly fabricated documents. Falsifying data means altering the existing records. For example undesired data or results are distorted or omitted. Plagiarism is an unacknowledged presentation or exploitation of work and ideas of others as one's own. Deception is the deliberate concealment of a conflict of interest. It includes deliberately misleading statements in research proposals or other documents.

4.5.4 Types of misconduct

Misconduct in clinical trial can be failure to follow an investigational plan, inadequate and inaccurate records, inadequate drug accountability, inadequate completion of informed consent forms, failure to report adverse drug reactions, failure to obtain document subject consent, failure to notify an Institutional Review Board or Ethics Committee of changes or progress reports, failure to obtain or document Institutional Review Board approval by Gupta [24].

4.5.5 Reasons why somebody commits fraud or misconduct

Reasons for fraud or misconduct in clinical trials are disparate from professional to personal. Fraud could be ambition like professional over to become famous, prestige being a part of international clinical trials or financial interests. Sometimes it could be laziness of the researcher or necessity repeat assessments for complex study. For example, repeating blood pressure measurements because blood pressure was rounded off to nearest 5 mm. Misconduct can happen when an investigator strongly believes intuitively in the "right" answer and does not respect the available evidence being contrary, due to ignorance or although due to oversight of the study. Misconduct can be backdating the subject's signature on a consent form because the subject forgot to date the form initially. Reasons for both include pressures for promotion and tenure, competition among investigators, ego, personality factors and conflicting personal and professional obligations. Existence of explicit versus implicit rules, penalties and rewards attached to such rules could be too reason for fraud or misconduct by Gupta [24].

4.5.6 Reasons why participants commit fraud or misconduct

Reasons why patients commit fraud or misconduct are various. Participant can be so interested in research that they can feel better or worst then their state of health is. But for example, it did not change the result of their blood tests. Bigger problem is when participants regularly cheat. The degree of cheating in one trial is a whopping 30 % by Marshall [25]. Participants forget to use medicine sometimes or they intentionally do not use medicine. Another type of cheating is dual enrolment into more then one clinical studies at one time by Barry [26]. All these types of cheating can change the result of clinical trial. Cheating in clinical trial is violation and can be classify like fraud or misconduct.

4.5.7 Impact of fraud and misconduct

The impact on affected individuals or research community could be significant. Fraud or misconduct can lead to repeating some aspects of research, which were fraudulent. Such incidents result in huge cost to the pharmaceutical company and also huge consequence for researchers. Disciplinary action can be lead with affected researchers or it may not be allowed them to be part of any advisory committee or peer review board. Articles publishes by such a researcher might be re-reviewed and retracted if required. Fraudulent clinical research affects the validity of data, what's more it affects rights, safety and well-being of research participants. In worth case, we would be able to buy ineffective or harmful molecules in the market by Gupta [24].

4.5.8 Detection of fraud and misconduct

Fraud or misconduct in clinical trials can be committed by all sites involved in clinical trial. There are a lot of inspecting mechanisms. Companies interested in clinical trial have their own mechanisms, even every country has owns regulations but every mechanism is aimed on another par of clinical fraud or misconduct.

Organization like Institutional Review Board and Ethics Committee should be active in strengthen research misconduct and fraud detection. They protect interest of research participants, simplification regulations and they made regulations more effective. They should have internal controls and review mechanisms for monitoring the ethical and quality aspects of ongoing studies.

One way how to detect fraud or misconduct is by data analysis. Data analysis can be done during the conduct of clinical trial. Warning signals can be excessive instances of perfect attendance on the scheduled day, 100% drug compliance, identical lab on electrocardiogram results, no serious adverse events reported or subjects adhering perfectly to a visit schedule [24]. Data analysis can be used like control mechanism by pharmaceutical companies as well as by government's organizations and institutions.

For example, FDA in USA is the most important in prevention and detection frauds in USA. If researchers have not compliance with the regulatory requirements or has engaged in fraudulent activity, then the FDA has the power to disqualify the investigator from taking part in further research [24].

4.5.9 Prevention of fraud and misconduct

Fraud and misconduct has many different forms as was explained above. Each form has different characteristic and should be solve particularly. Every pharmaceutical company has to solve this problem but they keep in secret detection process of fraud or misconduct. Also every country has own regulations of clinical trials to prevent research fraud. Generally it is impossible to prevent all fraud and misconduct that can be in clinical trial [24].

- "Adopt zero tolerance-all suspected misconduct must be reported and all allegations must be thoroughly and fairly investigated."
- "Protect whistle-blowers-careful attention must be paid to the creation and dissemination of measures to protect whistleblowers."
- "Clarify how to report-establish clear policies, procedures and guidelines related to misconduct and responsible conduct."
- "Train the mentors-researchers must be educated to pay more attention to how they work with their junior team members."
- "Use alternative mechanisms-institutions need continuing mechanisms to review and evaluate the research and training environment of their institution, such as internal auditing of research records."
- "Model ethical behavior-institutions successfully stop cheating when they have leaders who communicate what is acceptable behavior, develop fair and appropriate procedures for handling misconduct cases, develop and promote ethical behavior and provide clear deterrents that are communicated."

4.5.10 FDA inspections

FDA conducts clinical investigator inspections to determine if the clinical investigators are conducting clinical studies in compliance with applicable statutory and regulatory

requirements. Clinical investigators are required FDA investigators to access, copy, and verify any records or reports made by the clinical investigator.

FDA conducts both announced and unannounced inspections of clinical investigator sites, typically under the following circumstances [27]

- "to verify the accuracy and reliability of data that has been submitted to the agency
- as a result of a complaint to the agency about the conduct of the study at a particular investigational site
- in response to sponsor concerns
- upon termination of the clinical site
- during ongoing clinical trials to provide real-time assessment of the investigator's conduct of the trial and protection of human subjects
- at the request of an FDA review division
- related to certain classes of investigational products that FDA has identified as products of special interest in its current work plan (i.e., targeted inspections based on current public health concerns)."

4.5.11 Conclusion

This clinical trial summary gives us an overview of the area where medical companies develop new drug and potential medicine. Even it gives us guideline how to create a model for planning inspection.

5 Formalization

In this chapter the models of inspection scheduling problem are described. This chapter is divided into two sections. The first section deals with model of budget division. The second section deals with inspection scheduling problem in one Phase of clinical trial.

5.1 Scheme of the clinical trial

The scheme of the clinical trial represents a distribution of the global budget into partial budgets of individual phases. The budget of individual Phases is used to finance inspections in these Phases.

Every Phase has a different specification. Every phase is specified by the number of control weeks in the Time period and has some critical moments and every moment is critical by a different way. The results of critical moments decide about the future of the tested drug. The decision can be rejection of the drug or continuation of the clinical trial. A wrong rejection of an applicable drug means a potential loss of profit for the company if the drug is effective.

The pharmaceutical company needs correct data for the correct decision of the future of the drug for profit maximizing. Correct data are provided by inspections which are expensive and the cost of one inspection is different for each Phase of clinical trial. For example, inspection in one control week in Phase I is cheaper than in Phase III because less patients participate in Phase I. The best case is, if every Phase of clinical trial would be absolutely covered by inspections but it is not always possible. Thus, the goal of the complex scheme is to show risks with different share of inspection. It will helps pharmaceutical company to divide global budget effectively.

The description of the complex scheme is shown in Figure 8.

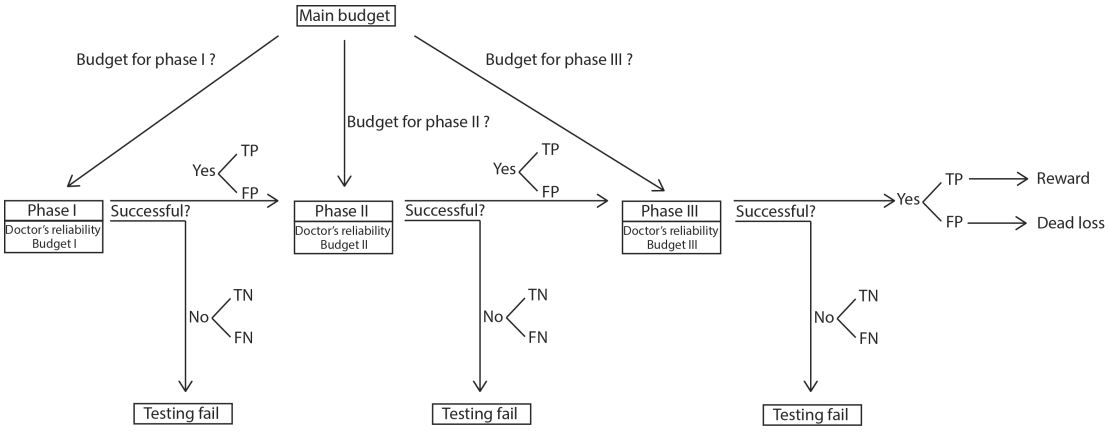


Figure 8 A scheme of dividing budget to three phases of clinical trial

5.1.1 Model of budget division

This model of budget division can be represented as a decision tree. The model of the decision tree focuses on the problem when a pharmaceutical company has a working drug in testing and another company has a similar drug. Then the other company may wants to thwart development of the drug in our pharmaceutical company. Briefly, they can bribe the doctor who works for our pharmaceutical company and then the doctor will want to conduct fraud by changing the result of testing. Thus, an inspector has to inspect the doctor every control week if the pharmaceutical company wants to absolutely know that the data are correct. But the pharmaceutical company has a limited budget for inspections and they have to divide budget in the most effective way. The decision tree can help them to decide about the budget because the decision tree shows them the decomposed problem of the budget division and potential risks. The decision tree for division budget into three Phases of clinical trial is shown in Figure 9.

Budget Division: Decision tree description

This decision tree is modeled for three different Phases of clinical trial. Every Phase has different parameters, such as number of control weeks in Time period, different importance of each week and a cost of one inspection. For example, Phase I can be defined with the following parameters: 3 control weeks in Time period, the importance of control weeks $\{0.5, 0.8, 1\}$ and the cost for inspection in one control week is 1. Phase II can be defined with the following parameters: 4 control weeks in Time period, importance of control weeks $\{0.9, 0.8, 0.7, 1\}$ and cost for inspection in one control week is 3. Phase III can be defined with the following parameters: 5 control weeks in Time period, importance of control weeks $\{0.6, 0.8, 0.9, 0.5, 1\}$ and cost for inspection in one control week is 10.

The pharmaceutical company faces the decision problem how much they have to inspect in each Phase. This decision problem is simplified to a problem with branching factor $b = 3$ (however, in reality, the branch factor is much larger). It means that the number of inspected weeks in each Phase can be 1/3 or 2/3 of weeks in the Time period or every week in the Time period.

If we have defined all previous parameters then values of leaves nodes have to be defined. Then the decision tree as is shown in Figure 9 can be then evaluated. This decision tree contains two types of nodes — G-nodes and Phase-nodes which are evaluated in a different way. G-node (uncertainty node) computes solution for one Phase using game-theoretic approach, as is described in following Section (5.2). Parameters for game depend on the type of Phase (i. e. Phase I, Phase II etc.) and the number of inspections. The simulation of the game with defined number of inspections gives us probability of following actions – FN (leaf node) and TP (node of following phase). FN represents the probability that the data from the Phase are changed by the doctor and TP represents that the data are correct. Phase-node represents decision node. Possible actions in decision node are different options how much the inspector can inspect the doctor. The pseudo-code of this algorithm is shown in Algorithm 1.

Algorithm of Decision tree description

The Algorithm (1) starts with function *buildTree* with the input node $n_c(1)$. The node n_c is initial node and his type is t_1^P. The type t represents a group of all possible types of nodes in the decision tree. Specifically, type t_a^P is type of Phase node and a represents the number of the Phase (i.e Phase I, Phase II etc.). Type t^G represents

Algorithm 1 The decision tree

```
 1: function buildTree(n_c)
 2:     switch n_c.getType() do
 3:         case t_1^P:  buildPhase(n_c, d_1, I_1, t_1^C, c_1)
 4:         case t_2^P:  buildPhase(n_c, d_2, I_2, t_2^C, c_2)
 5:         case t_3^P:  buildPhase(n_c, d_3, I_3, t_3^C, c_3)
 6:         case t_1^G:  buildGame(n_c, d_1, P_1, t_2^P)
 7:         case t_2^G:  buildGame(n_c, d_2, P_2, t_3^P)
 8:         case t_3^G:  buildGame(n_c, d_3, P_3, t^L)
 9:         case t^L   n_c.Value ← 100000
10: end function
11:
12: function buildGame(n_c, d, P, t)
13:     p ← SSE(d, n_c.getInspections, P)
14:     n_FN ← addFNNode(n_c, p.getFN, value-0, t^L)
15:     n_TP ← addTPNode(n_c, p.getTP, t)
16:     n_TP ← buildTree(n_TP)
17:     n_c.Value ← n_TP.getValue · n_TP.getProbability
18: end function
19:
20: function buildPhase(n_c, I, t, c)
21:     for all i ∈ I do
22:         cost ← i · c + n_c.getPreviousCost
23:         if  cost ≤ maxBudget then
24:             n_ch ← buildTree(new Node(n_c, t, cost, i))
25:             n_c.addChildren(n_ch)
26:         end if
27:     end for
28: end function
```

Game and type t^L represents Leaf. Number of control weeks in Time period for one Phase is represented in algorithm as d_a where a is index of corresponding Phase. The I_a represents group of choices how to inspect in Phase a and the cost of one inspection in Phase a is represented as c_a. Set of weights for weeks in the Time period in Phase a is stored in P_a. Values of d_a, I_a, c_a, P_a, with a ranging from 1 to 3 and maxBudget are given by default.

As was previously stated, the Algorithm (1) starts in function *buildTree* (1) which defines the decision tree level by level and the function recursively builds the Decision tree. Because the Decision tree contains two types of internal nodes—Phase node and Game node, the function *buildTree* builds the Decision tree with help of two other functions.

The function *buildPhase* (20) has one of the input arguments node n_c. This function generates possible children of the node n_c (24, 25). The type of the node n_c corresponds to the Phase node which is represented as a decision node. The set of possible decisions is set of choices how to inspect current Phase under constraint that the cost of inspections in this Phase and previous Phases is equal or lower than the maximal budget (23). For example, if the Time period contains three weeks and the maximal budget is enough for three inspections then possible decisions are inspect one, two or three weeks in the

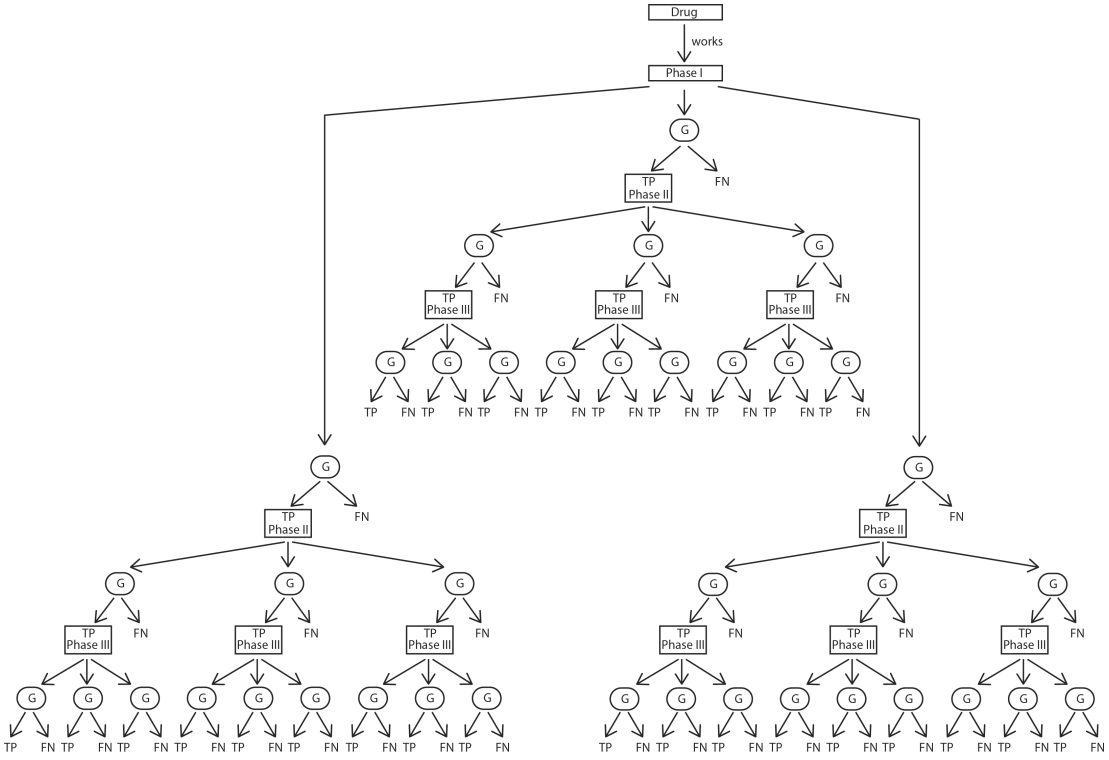


Figure 9 Decision tree for budget division

Time period. The Decision tree is recursively build with all possible children node of input node n_c (24).

The function *buildGame* (12) plays game of the previous Phase with the given number of inspections (see Section 5.2). Then this function generates two children nodes for the input node n_c (14,15). The first child node is a leaf node n_{FN} (14) which represents option that doctor changed the result of the testing and the inspector does not know about it. The value of this node is equal to zero. The second child node is positive node n_{TP} (15) which contains probability that the inspector correctly inspects data. The decision tree is recursively built with the node n_{TP} (16).

5.2 Formalization of the inspection scheduling problem in one Phase

5.2.1 Problem definition

The model with formalization of the inspection scheduling problem in one Phase represents the interaction between the doctor and the inspector in a single Time period of Phase.

Time period P is the duration of one phase of the clinical trial. It is represented as a number of control weeks in which the patients are controlled by the doctor and the number of control weeks depends on the type of the drug and type of the Phase.

Doctor controls patients in the control weeks. The doctor wants to perform fraud in this type of model. But he prefers not to be revealed by the inspections. The doctor can be only one person if the trial is small. If the trial is bigger and it needs more doctors then the doctor represents group of doctors who want to do fraud.

Inspector is a member of pharmaceutical company and he is the person who is responsible for detecting and preventing frauds. Every Phase has a given budget for inspections. The size of the budget is represented as the maximum number of inspections which inspector can realize in the Time period. He leads inspections and if he inspects doctor who cheats, he detects the fraud.

The goal of our model is to find the schedule of inspections for inspector which will maximize the probability of detecting doctor's fraud our deter the doctor from conduct fraud altogether.

5.2.2 Decomposition of type of games with different complexity

In this section, models of the two-person security games with different complexity are analyzed. In these games, one agent is defender and the second agent is attacker. In model of inspection problem, the defender is inspector and the attacker is doctor. Defender wants to protect attacker's targets from attacker's attacks. Models could have parameters like zero-sum vs. non-zero sum, simultaneous move vs. leader-follower and types of models are shown in following table.

	zero-sum	non-zero sum
simultaneous move	Game 1	Game 2
leader-follower	Game 3	Game 4

Decomposition of Game 1 and Game 3

Both these games are zero-sum games. Nash equilibrium is used for solving Game 1 where both agents play simultaneously. Strong Stackelberg equilibrium is used for Game 2 where defender is the leader and the attacker is the follower. Defender has to suppose that attacker will obtain leader's strategy.

In some situations follower may chooses to act without acquiring leader's security strategy. Especially, if the security measures are difficult to observe. Then the leader faces an unclear choice about which game to play in this case Game 1 or Game 3. Relationship between the NE in Game 1 and SSE in Game 3 strategies in security games is following. For finite two-person zero-sum security games, it is known that game theoretic solution concepts of NE and SSE give the same strategy [4].

In general settings, the equilibrium strategy can in fact differ between the game with NE and SSE. But the Nash equilibrium strategies of zero-sum games have a property in that they are interchangeable [28].

Decomposition of Game 2 and Game 4

Both these games are non-zero sum and they are derived from previous Games 1 and 2. The typical solution concept applied to Game 2 is Nash equilibrium and Strong Stackelberg for Game 4.

Defender faces the same problem, which game to play. But answer in non-zero sum games is not as straight as in zero-sum games. However, if the non-zero-sum game is security game (2.4.1) and satisfies the SSAS (Subset of Schedules Are Schedules) property, then the defender's set of SSE strategies is a subset of his NE strategies [4].

In conclusion, the zero-sum game using Nash Equilibrium, however, we solve the non-zero-sum game using Strong Stackelberg Equilibrium, which is more realistic in real-world inspection scheduling problems.

5.2.3 Security game of the inspection scheduling problem in one Phase

Security game of the inspection scheduling problem in one Phase contains two agents, doctor and inspector. The inspector is defender of the game and the doctor is attacker. Both agents have set of strategies which can play.

The doctor is player who wants to perform fraud. He knows that he is checked by the inspector but he knows only probability of the inspections for every control week in the Time period. The doctor's strategies contain two types of weeks. The first type is week when the doctor correctly works and the second type is week when doctor performs fraud. So, the doctor's strategies are permutation with repetition of two-choices (cheating week and normal week), which are repeated r-times and r correspond to number of control weeks in the Time period.

The inspector is player who has profit when he detects or prevents doctor's fraud. The inspector has given number of inspections and this number of inspections corresponds to the given budget for inspections in one Phase. Inspector's strategies are permutation of vector, which size is equal to number of control weeks in the Time period and which contain on x positions 1 where x is equal to number of inspections and other positions contain 0 which represent any inspection in the week.

There are examples of doctor's T and inspector's S strategies. Let's imagine the following situation, the Time period contains three control weeks and the inspector can inspect in two control weeks in the Time period. Then the set of inspector's strategies is $S = \{110, 011, 101\}$ where 1 state represents inspection and 0 state represents no inspection in corresponding control week. The doctor's strategies are $T = \{000, 100, 010, 001, 011, 101, 110, 111\}$ where 1 state represents fraud and 0 represents no fraud in corresponding week.

The doctor has a pay-off matrix based on doctor's utility function, where every doctor's strategy is evaluated with every inspector's strategy. The inspector has a pay-off matrix based on inspector's utility function, where every inspector strategy is evaluated with every doctor's strategy.

5.3 Utility models

5.3.1 Time-independent utility model

Model time-independent works on game described in Section (5.2.3). The model represents the inspector as a leader and defender in the game and the doctor is a follower and attacker.

The inspector wants to find the best commit to mixed strategy which will represents his effort to detect and prevent doctor's fraud. The doctor wants to perform fraud but he does not want to be detected by the inspector. The doctor knows the leader's mix strategy and he wants to play the best strategy to the leader's strategy.

In this model, all weeks in the Time period has the same importance. Both players in this model want to maximize their utility function.

Doctor's utility function

Doctor's utility function U_D reflects the type of doctor who is inspected. Without inspections the doctor is strongly inclined to perform a fraud. Another type is doctor who does not want to perform fraud.

In this type of game is expected that the doctor wants to perform fraud because he expects some reward. Thus, if the doctor performs fraud only in one control week in the

Time period his reward is much lower than if the doctor performs fraud every control week in the Time period but even the risk is bigger.

When the doctor's fraud is detected by the inspector then the doctor expects a punishment. The doctor knows what the punishment is. For example, the doctor has to pay a penalty to the pharmaceutical company. This fact is written in the doctor's agreement with the pharmaceutical company. Thus, if the doctor knows about the punishment and he still wants to do fraud, his reward has to be higher than the punishment. In another way, the doctor would not be enough motivated to do fraud.

The doctor is rational and he wants to maximize his utility function U_D. The doctor's utility function can be defined as sum of values of each doctor's control weeks in the Time period as in Equation (5). The value of each week is represented in the following Equation (4). Where P is Time period, d is one week of Time period, v_d^D is value of single week d for the doctor, t is doctor's pure strategy, s is inspector's pure strategy and U_D is doctor's utility function.

$$v_d^D = \begin{cases} loss & \text{if } t_d\text{=cheat and } s_d\text{= inspection} \\ profit & \text{else if } t_d\text{=cheat and } s_d\text{= no inspection} \\ 0 & \text{else} \end{cases} \tag{4}$$

$$U_D = \sum_{d \in P} v_d^D \tag{5}$$

Inspector's utility function

The inspector's utility function U_I reflects inspector's mission. He is a member of the pharmaceutical company and he is responsible for the correct data from the testing.

His inspections have to detect fraud if the doctor performs fraud and have to have a prevention effect. Thus, the inspector fails in his function, when he does not detect the doctor's fraud in any week in the Time period and the doctor performs fraud in some weeks in the Time period. It means that his schedule for inspecting the doctor was not good. If the doctor does not perform fraud in any control week in the Time period, the inspector gets the same reward as if he detects doctor's fraud. Because when he inspects and the doctor does not perform fraud it can be due to the prevention effect of his inspections and it means, that schedule for inspections is good.

We can represent inspector's utility function as equation (8). Where P is Time period, d is one week of Time period, t is doctor's pure strategy, s is inspector's pure strategy and U_I is inspector's utility function. Inspector's week d when the inspector does not inspect and the doctor cheats is captured by the v_d^{NI} variable. Inspector's week d when the inspector inspects and the doctor cheats is captured by the v_d^I variable.

$$v_d^{NI} = \begin{cases} 1 & \text{if } t_d\text{=cheat and } s_d\text{= no inspection} \\ 0 & \text{else} \end{cases} \tag{6}$$

$$v_d^I = \begin{cases} 1 & \text{if } t_d\text{=cheat and } s_d\text{= inspection} \\ 0 & \text{else} \end{cases} \tag{7}$$

$$U_I = \begin{cases} 1 & \text{if } \sum_{d \in P} v_d^I > 0 \\ 0 & \text{else if } \sum_{d \in P} v_d^{NI} > 0 \\ 1 & \text{else} \end{cases} \tag{8}$$

Example of utility functions

Let's imagine the following situation. The time period has three weeks and the inspector will inspect only one control week in the Time period. The doctor's profit is 60 and the doctor's loss is -40 for counting his utility function.

For example, the doctor's strategy is $t = \{011\}$ and inspector's strategy is $s = \{001\}$. Thus, doctor will do fraud at the second and at the third week and inspector will inspect the third week.

Then doctor's utility function $U_D(t, s)$ is sum of $v_{d=1}^D = 0$, $v_{d=2}^D = 60$, and $v_{d=3}^D = -40$. Therefore, doctor's utility function is $U_D(t, s) = 20$.

Inspector's utility function $U_I(t, s)$ work on $v_{d=1}^I = 0$, $v_{d=2}^I = 0$, and $v_{d=3}^I = 1$. Thus, inspector's utility function is $U_I(t, s) = 1$.

5.3.2 Time-dependent utility model

Doctor model time-dependent is derived from the Doctor model time-independent. The main difference is in the importance of control weeks in the Time period. Some control weeks are more critical. The importance of weeks in the Time period depends on the type of the drug and the Phase.

For example, the first control week is very important, because we will compare state of health before the drug starts to effect and after. Thus, the effect of the drug would be the most dynamic in the middle of the Time period and the result of these weeks are really important. But some weeks can be only control weeks for patients, if they really use the drug correctly and if the patients have the right amount of drug in their blood. The result is that some weeks are more important in decision making of the future of the drug.

The doctor and the inspector have experience with the clinical testing and they both know which weeks are important in decision making of the future of the drug.

The different importance of control weeks changes the utility functions from the previous model.

Doctor's utility function

Doctor's utility function U_D is derived from Doctor's utility function in Section (5.3.1). The doctor has the same goal but he knows the importance of the control weeks in the Time period.

As before, the reward gives him somebody who wants to change the results of testing as much as possible. Thus, if doctor cheats in less important control weeks in the Time period, the reward is lower than if the doctor cheats in more important control weeks in the Time period.

The doctor is rational and he wants to maximize his utility function. The doctor's utility function can be defined as the sum of values of every doctor's control week in

Time period as is shown in equation (10). Where P is Time period, d is one week in the Time period, v_d^D is value of single week d for the doctor, t is doctor's pure strategy, s is inspector's pure strategy and U_D is doctor's utility function.

$$v_d^D = \begin{cases} loss & \text{if } t_d\text{=cheat and } s_d\text{= inspection} \\ profit \cdot i^d & \text{else if } t_d\text{=cheat and } s_d\text{= no inspection} \\ 0 & \text{else} \end{cases} \tag{9}$$

$$U_D = \sum_{d \in P} v_d^D \tag{10}$$

Inspector's utility function

The inspector's utility function U_I is derived from the inspector's utility function in Section (5.3.1). Inspector has the same goal but he knows the importance of the control weeks in the Time period.

Inspector's utility function is represented in equation (14). Where P is the Time period, d is one control week in the Time period, t is doctor's pure strategy, s is inspector's pure strategy, U_I is inspector's utility function and α represents weights of control weeks in the Time period. Inspector's week d when the inspector does not inspect and the doctor cheats is captured by the v_d^{NI} variable. Inspector's week d when the inspector inspects and the doctor cheats is captured by the v_I^d variable.

$$v_d^{NI} = \begin{cases} 1 & \text{if t}(d)\text{=cheat and s}(d)\text{= no inspection} \\ 0 & \text{else} \end{cases} \tag{11}$$

$$v_d^I = \begin{cases} 1 & \text{if t}(d)\text{=cheat and s}(d)\text{= inspection} \\ 0 & \text{else} \end{cases} \tag{12}$$

$$\alpha_{max} = max\{\alpha^d\} \quad (\forall d \in P \mid t(d) = \text{cheat } \& \ s(d) = \text{inspection}) \tag{13}$$

$$U_I = \begin{cases} 1 \cdot \alpha_{max} & \text{if} \sum_{d \in P} v_d^I > 0 \\ 0 & \text{else if} \sum_{d \in P} v_d^{NI} > 0 \\ 1 & \text{else} \end{cases} \tag{14}$$

Example of utility functions

Let's imagine the following situation. The Time period has three weeks and inspector will inspect only in one control week in the Time period. Importance of 1^{st} *week* is 0.2, of 2^{nd} *week* is 0.7 and of 3^{rd} *week* is 0.9. Doctor's profit is 60 and doctor's loss is -40 for counting doctor's utility function.

For example, doctor's strategy is $t = \{011\}$ and inspector's strategy is $s = \{001\}$. Thus, doctor will do fraud at the second and at the third week and inspector will inspect

the third week.

Then doctor's utility function $U_D(t, s)$ is sum of
$v^D_{d=1} = 0$, $v^D_{d=2} = 60 \cdot 0.7$, and $v^D_{d=3} = -40$. Therefore, doctor's utility function is $U_D(t, s) = 2$.

Inspector's utility function $U_I(t, s)$ work on $v^I_{d=1} = 0$, $v^I_{d=2} = 0$, and $v^I_{d=3} = 1$. And α_{max} is $\alpha_{max} = max\{\alpha^{d=3}\} = max\{0.9\}$. Thus, inspector's utility function is $U_I(t, s) = 0.9$.

Conclusion

The utility models described in this section are used for non-zero sum game, which is counted by Strong Stackelberg equilibrium. The goal of this game is to find for every control week in the Time period probability that the inspector will inspect the doctor.

5.3.3 Zero-sum game approximation

In this subsection, zero-sum game approximation is presented. This game contains two agents—doctor and inspector. Doctor's utility function U_D reflects the incentive of a doctor who wants to do fraud and without an inspection he is strongly inclined to do so. Thus, the doctor's utility function is represented in the following Equation (17). Where P is Time period, d is one week of Time period, t is doctor's pure strategy and s is inspector's pure strategy and. Label of the inspector's week d when the inspector does not inspect and the doctor cheats is v^{NI}_d. Inspector's week d when the inspector inspect and the doctor cheats is captured by the v^I_d variable.

$$v^{NI}_d = \begin{cases} 1 & \text{if t}(d)\text{=cheat and s}(d)\text{= no inspection} \\ 0 & \text{else} \end{cases} \tag{15}$$

$$v^I_d = \begin{cases} 1 & \text{if t}(d)\text{=cheat and s}(d)\text{= inspection} \\ 0 & \text{else} \end{cases} \tag{16}$$

$$U_D = \begin{cases} -1 & \text{if } \sum_{d \in P} v^I_d > 0 \\ 1 & \text{else if } \sum_{d \in P} v^{NI}_d > 0 \\ 0 & \text{else} \end{cases} \tag{17}$$

Then, the doctor's utility function is defined as $U_I = -U_D$.
For this type of game the solution is found by Nash Equilibrium.

6 Solution

Practical implementation of solution concepts for solving inspection planning problem was implemented in Java SE 8. These concepts contain LP problems which were solved by IBM solver CPLEX.

6.1 Solution of Security game of the inspection problem in one Phase

The game contain two agents—doctor and inspector.

The values of every pair of inspector's strategy and doctor's strategy is stored in three-dimensional matrix $A_{m,n,o}$ where each i^{th} row correspond to i^{th} doctor's strategy and j^{th} column correspond to j^{th} inspector's strategy. Each cell has values given by utility functions of pair of i^{th} doctor's strategy and j^{th} inspector strategy for doctor $U_D(i,j)$ in matrix $A_{m,n,2}$ and for inspector $U_I(i,j)$ in matrix $A_{m,n,1}$.

$$A_{m,n,1} = \begin{pmatrix} U_I(1,1) & \cdots & U_I(1,n) \\ \vdots & \ddots & \vdots \\ U_I(m,1) & \cdots & U_I(m,n) \end{pmatrix}$$

$$A_{m,n,2} = \begin{pmatrix} U_D(1,1) & \cdots & U_D(1,n) \\ \vdots & \ddots & \vdots \\ U_D(m,1) & \cdots & U_D(m,n) \end{pmatrix}$$

This representation is implemented in class *Matrix*.

6.1.1 Solution for inspection planning problem computed by NE

First, we solve computationally easier zero-sum game with opposite utilities (see Section 5.3.3), where both agents plays simultaneously. The doctor has the set of pure strategies T and the inspector has the set of strategies S. This game is implemented in class *InspectionProblemNash*.

The solution of this game is found by linear program (18) for computing Nash equilibrium [1]. Variables in this linear program are mixed strategy terms p_s and v. This linear program gives us inspector's mixed strategy in equilibrium.

$$\begin{aligned} & \min v \\ & \text{subject to} \\ \forall t \in T \quad & \sum_{s \in S} p_s \cdot U_D(s,t) \leq v \\ & \sum_{s \in S} p_s = 1 \\ \forall s \in S \quad & p_s \geq 0 \end{aligned} \tag{18}$$

6.1.2 Solution for inspection planning problem computed by SSE

Inspection planning problem is represented as security game with to commit to strategy. This problem is implemented in class *PlaningInspectorScheduleProblem*.

As was previously stated, the game has two players — doctor and inspector. The doctor has the set of pure strategies T and inspector has set of pure strategies S.

Then SSE can be computed as presented by Conitzer and Sandholm et al [2]. For every pure doctor strategy t a mixed strategy for inspector is computed while assuming t is a best response for the doctor's mixed strategy. The SSE can be computed using the following linear programs (19). Variable p_s is a probability of s^{th} inspector strategy. Linear programs are solved with the CPLEX solver.

$$\begin{aligned} \forall t \in T \max \quad & \sum_{s \in S} p_s \cdot U_I(s,t) \\ \text{subject to} & \\ \forall t' \in T \quad & \sum_{s \in S} p_s \cdot U_D(s,t) \geq \sum_{s \in S} p_s \cdot U_D(s,t') \\ & \sum_{s \in S} p_s = 1 \end{aligned} \tag{19}$$

If the linear programs are solved with a single best solution for the inspector then the inspector knows probability for each of his strategy. If the linear programs are solved with more than one possible solution then the inspector has set of mixed strategies which has for the inspector the same value computed by inspector's utility function.

6.2 Implementation

As it has been mentioned, the the algorithms for compoting Nash and SSE were implemented in Java SE.

The utility functions described in Section (5.2.3) are implemented in class *UtilityFDTimeDep* for the doctor and in class *UtilityLCTimeDep* for the inspector. The values of the utility functions for pairs of doctor and inspector strategy are stored in three-dimensional matrix which is implemented in class *Matrix*.

The solution for inspection schedule problem solved be NE described in Subsection (6.1.1) is implemented in class *InspectionProblemNash*. The solution for the inspection schedule problem solved by SSE described in Section (6.1.2) is implemented in class *PlaningInspectorScheduleProblem*. The concept for finding SSE solution for inspection schedule problem from Section (6.1.2) is computed as the set of linear programs, where each linear program is solved by the IBM CPLEX solver. Some of these linear programs may be in-feasible if these programs are solved for some doctor strategies t. For example, if t is a doctor's strictly dominated strategy (see Section 2.4).

The decision tree for budget division is implemented in class *Tree* and the j-graph library is used for the visualization this decision tree.

7 Evaluation

In this chapter, we evaluate algorithms proposed above. This chapter is divided into two sections. The first section deals with the evaluations of inspection scheduling problem. We focus on evaluation SSE, because NE is less computationally hard and it is less realistic. The second section deals with the problem of budget division.

All test in this chapter were performed on synthetic data.

7.1 Deployment to saturation ratio

In this section, we will focus on the runtime required by the algorithm with different parameters and finding the hardest combination of them. In this case, the parameters are number of weeks in the Time period and number of inspections. The number of weeks must be minimally equal or higher than number of inspections.

The evaluation in this section is based on the concept of deployment to saturation $(d : s)$ ratio. The concept shows that the problem exhibits a phase transition at 0.5 for random Stackelberg Security Game instances [29], and shows that the hardest instances arise at this point. The $(d : s)$ ratio has the following definition: the deployment refers to the number of defender's resources available to be allocated, and the saturation refers to the minimum number of defender's resources such that the addition of further resources beyond this point yields no increase in the defender's expected utility. In this case the $(d : s)$ ratio is represented as the number of inspections divided by the number of weeks in the Time period.

Experiments were executed on Intel i7 processor with 16 GB RAM and CPLEX 12.5 was used as the LP solver. The evaluation was executed for number of control weeks ranging from 1 to 9 with a random value of each day in the Time period. The evaluation

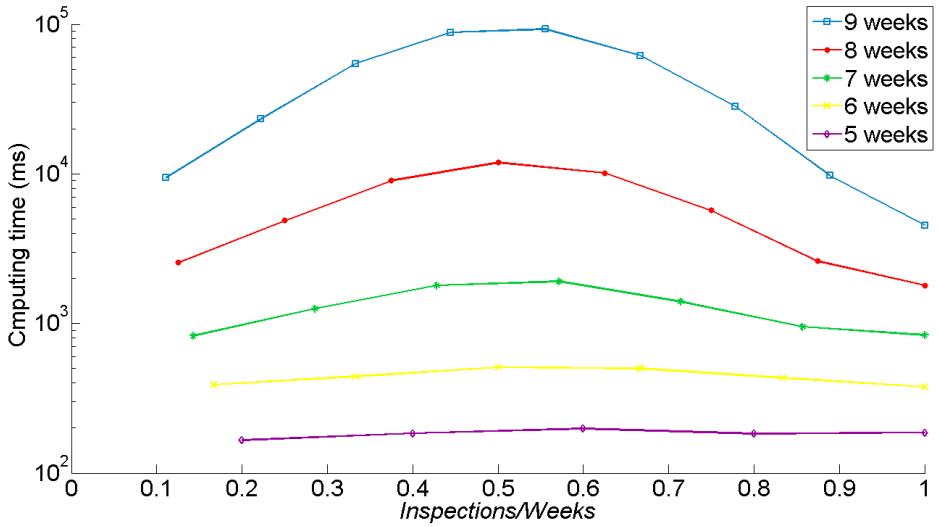


Figure 10 Deployment to saturation ratio

for number of weeks less than five in the Time period has very similar computation time for different number of inspection. In this evaluation we focus to find the hardest combination of input parameters. Thus, we focus on the number of weeks higher than 4.

The scenarios for the evaluation had the initial number of control weeks in the Time period ranging from 5 to 9. For each scenario the test was executed for number of inspections ranging from 1 to i, where i refers to the number of control weeks in the Time period. Every test of scenario was executed 10 times. The average results of tests are depicted in Figure 10. The x-axis contain $(d : s)$ ratio, it is number of inspections divided by the number of control weeks in the Time period. The y-axis represents the runtime in milliseconds and it has a logarithmic scale.

We can see the expected result of the graph in the Figure 10. The worst cases of input argument are those around $(d : s) \approx 0.5$. On the other hand cases with minimal and maximal $(d : s)$ ratio are the easiest to compute. This result is expectable because it matches sizes of pay-off matrix which are the largest around $(d : s) \approx 0.5$ than in other values of the $(d : s)$ ratio.

7.2 Comparison inspector's strategy computed by SSE with other types of strategies

Inspections can be planned by many different ways. In this section we compare strategy computed by Strong Stackelberg equilibrium with greedy strategy and uniform strategy.

Strategies are planned for scenario with eight control weeks in the Time period and four inspections. The importance of the control weeks in the Time period is shown in Figure 11. The first day is very important because doctor measures patients health before the drug will have an effect. Than the drug is the most effective at the fourth and the fifth control day. The doctor can measure effectiveness of the drug and results are very important for the pharmaceutical company. Next control weeks will not show so much about effectiveness of the drug but the last day is important for the next comparison with previous results of the testing. This fact about day importance is known to the doctor even as the inspector.

In these evaluation, we expect the doctor is motivated to do fraud, thus the doctor's utility function will be computed with the loss equal to -40 and with the profit equal to 60.

7.2.1 Strategy computed by strong Stackelberg equilibrium

The best strategy for the inspector computed by a Strong Stackelberg equilibrium is shown in Figure 12. The inspector will commit to this strategy and the doctor's best response to the inspector's strategy is shown under the inspector strategy in Figure 12. Then value of inspector strategy is equal to 1 and value of doctor strategy is equal to 19.9.

7.2.2 Greedy strategy

The inspector can also choose not to compute the SSE, however, use a more simplistic greedy strategy. The inspector easily chooses the most important weeks in the Time period and he will only inspect the most important weeks. The doctor will respond to the inspector's strategy and the doctor will perform fraud in the weeks when the inspector will not inspect as it is shown in Figure 13.

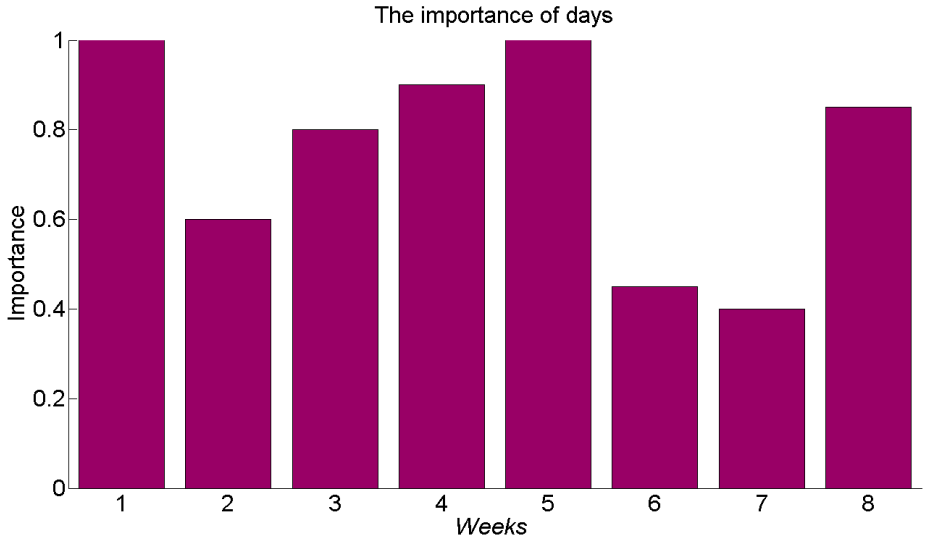


Figure 11 The importance of control weeks in the Time period

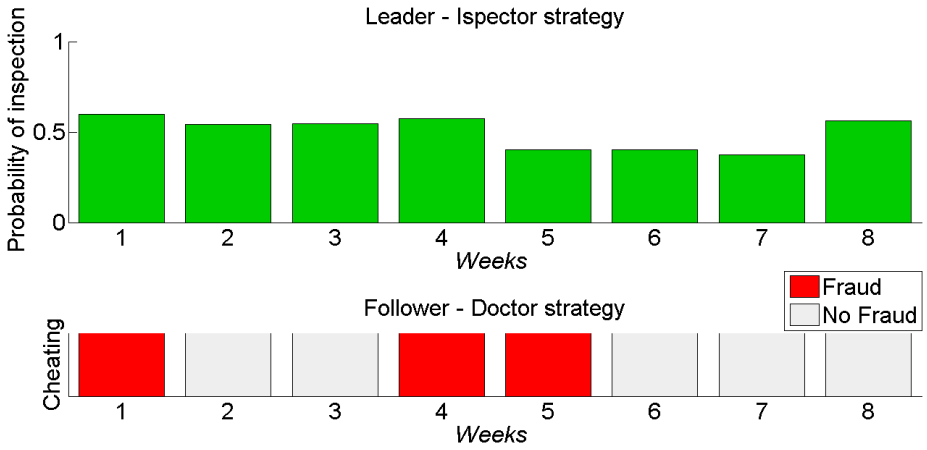


Figure 12 SSE Strategy with 8 control weeks in the Time period and 4 inspections

Thus we can see that in some weeks the doctor can easily cheat and he knows that inspector will not inspect him. Even if the doctor does not cheat in the most important weeks, he can change the result of the trial. So, this type of inspector's strategy does not have any preventive effect and value of inspector strategy is equal to just 0.0 and value of doctor strategy is equal to 135.

7.2.3 Uniform strategy

Another strategy for the inspector is uniform strategy. The inspector's strategy will have the same probability of inspection in every control week of the Time period as it is shown in Figure 14. Then the doctor's best response is cheat every week and value of inspector strategy is 0.9 and value of doctor strategy is 36.5.

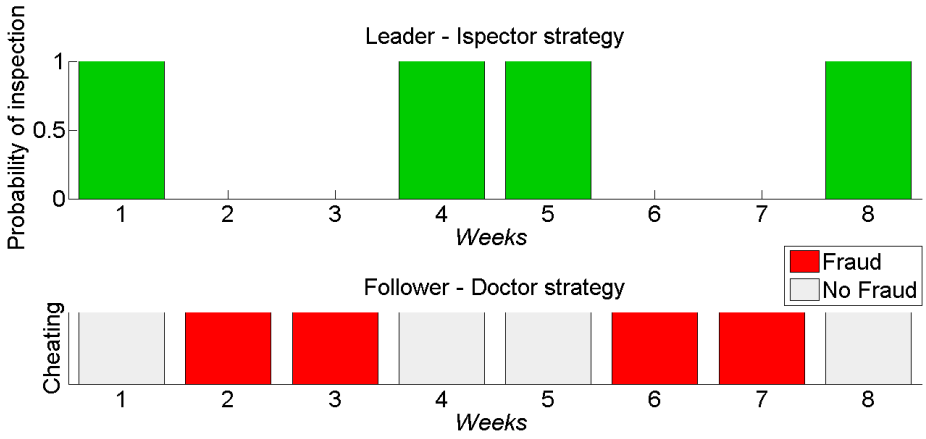


Figure 13 Greedy strategy with 8 control weeks in the Time period and 4 inspections

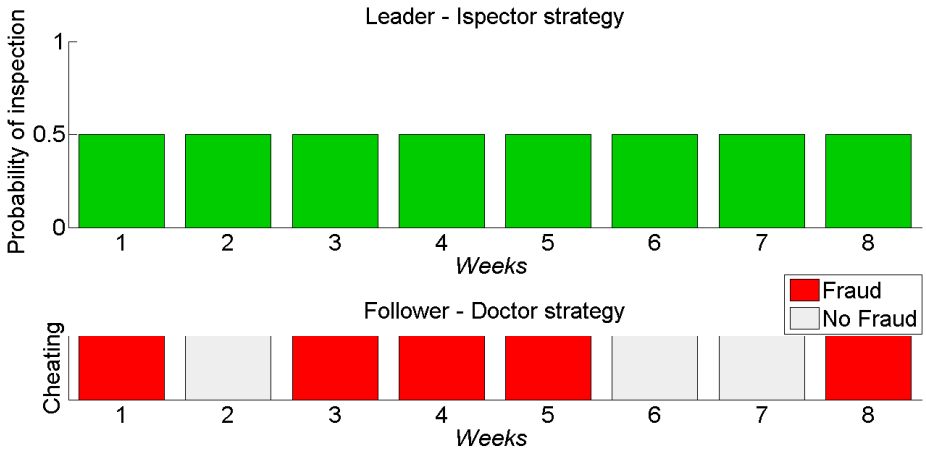


Figure 14 Uniform strategy with 8 control weeks in the Time period and 4 inspections

7.2.4 Summary

All strategies presented above are possible for inspector. But if we compare them by the value of the inspector's strategy the worst is greedy strategy. If the doctor is clever and he wants to perform fraud, it is really easy for him to change completely the results of the Phase in the less important control weeks. Additionally, he can cheat and knows that he will not be inspected. Greedy strategy is very bad strategy with no prevention effect in comparison with other types of inspector's strategies.

Uniform strategy has a higher value of inspector strategy than greedy strategy. The value of inspector's strategy is greater than the value of the greedy strategy, but if we focus on the doctor's best response to the inspector strategy, the doctor cheats in the the most important weeks and in more then half weeks in the Time period. Even if inspector's strategy covers weeks when the doctor performs fraud but the strategy does not cover these days optimally. Thus, we can see that this inspector strategy has not

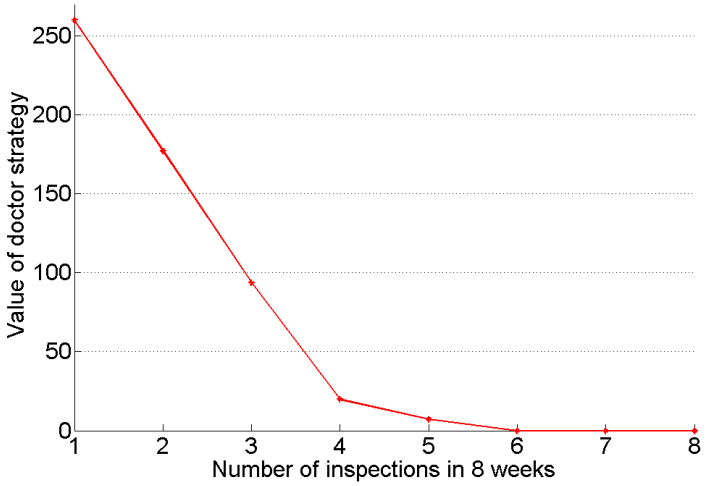


Figure 15 Dependence values of doctor's strategies on number of inspection in 8 weeks in the Time period

enough prevention effect. Moreover, inspector's strategy motivates doctor to cheat in the most important weeks and change the result of the Phase.

The highest value of inspector strategy has a strategy computed by the Strong Stackelberg equilibrium. We can see that the best response for the doctor is to cheat only in three weeks. Thus, inspector's strategy computed by the SSE covers the weeks in the Time period by the most optimal way for a given number of inspections and because the doctor is strongly motivated to perform fraud the four inspections are not enough to demotivate the doctor to perform fraud. This inspector strategy has bigger prevention effect than inspector's strategies computed as greedy strategy or uniform strategy. Thus, the best strategy from these three types of inspector's strategies is strategy computed by SSE.

Inspector's strategy type	Value inspector's strategy	Value of doctor's strategy
SSE	1	19.9
Greedy strategy	0.0	135
Uniform strategy	0.9	36.5

7.3 Incentives for the doctor to perform fraud

If the pharmaceutical company has not budget to cover every control week in the Time period by the inspection, then it is useful to know how the doctor reacts to different numbers of inspections. Thus, the dependence of the value of doctor's strategy on the number of inspections will be discussed in this section.

We use a similar scenario as in the previous section. Specifically, the Time period contains 8 control weeks with importance of weeks as is shown in Figure 11. Doctor is motivated to do fraud and for counting his utility function is used loss equal -40 and profit equal 60 and numbers of inspections will increase from 1 to 8. We will focus on the dependence of doctor's motivation to perform fraud on the number of inspections which is shown in Figure 15.

We can see that for one inspection in the Time period the value of the best doctor

Figure 16 Comparison strategy with four inspection with strategy with six inspections

strategy is equal to 260. Then this value steeply decreases with increasing number of inspections to four inspections where the value of doctor's best strategy is equal to 19.9. Then value of doctor's strategy slowly decreases to six inspections where the value of doctor's strategy is equal to 0.

Thus, if the pharmaceutical company in this case uses four inspections to control the doctor, then they significantly reduce doctor's motivation to perform fraud. If the pharmaceutical company wants to demotivate the doctor to perform fraud, they have to have budget minimally for six inspections. For better idea, comparison of the doctor and the inspector strategies for four and six inspection's weeks in the Time period is shown in Figure 16. We can see that the inspector's strategy for six inspections in the Time period covers the important weeks when the doctor is motivated to perform fraud. The inspector strategy for four inspections in the Time period reduce the doctor's motivation to perform fraud but this number of inspections is not enough to demotivated the doctor to perform fraud.

In conclusion, the doctor will not demotivated to perform fraud if the pharmaceutical company will cover the Time period with six or more inspections.

7.4 Budget division

If the pharmaceutical company has a limited budget or even if they want to inspect Phases of testing optimally then the pharmaceutical company wants to know risks and the optimal solution how to inspect in each Phase of the clinical trial. The model of budget division which is described in Subsection (5.1.1) as a decision tree is evaluated in this section with the following Scenario for budget division.

The scenario for budget division contains three Phases of clinical trial. Phase I is defined with the following parameters: three control weeks in Time period, importance of weeks in the Time period is $\{0.5, 0.8, 0.1\}$ and cost for inspection in one week is 1. Phase II is defined with the following parameters: four weeks in the Time period, importance of weeks is $\{0.9, 0.8, 0.7, 1\}$ and cost for one inspection is 3. Phase III can be defined with the following parameters: five weeks in the Time period, importance of weeks is $\{0.6, 0.8, 0.9, 0.5, 1\}$ and cost for inspection in one control week is 10. Options how to inspect Phase I are inspected one, two or three weeks in the Time period.

Options how to inspect Phase II are inspected two, three or four weeks in the Time period. And options how to inspect Phase III are inspected two, four or five weeks in the Time period.

Firstly, Scenario of budget division is evaluated with budget equal to 65. This budget is enough to cover with inspections every control week in every Phase of clinical trial. The resulting decision tree is shown in Figure 17.

We can see that the optimal number of inspections in Phase I is two, for Phase II it is three inspections and for Phase III it is four inspections. Thus, if the Phases are optimally inspected then they do not be covered have to be the inspection for every week to guarantee correct data.

Second, scenario of budget division is evaluated with the limited budget which is equal to 27. The result of evaluation is shown in Figure (18). We can see that Phase I is covered with two inspections Phase II is inspected only with two inspections and Phase III is inspected only with two inspections. Previously, Phase II was inspected with three inspections and Phase III with 4 inspections.

Hence, this scenario shows that it is possible to optimally divide the inspections for limited budget, but the uncertainty that pharmaceutical company receive incorrect data is higher than for a higher global budget. The uncertainty, that this division of global budget does not guarantee the correct data is taken into consideration in value of Phase 1 node, which has lower value than in division for higher budget.

For a better understanding, Figures of Budget division, blue node P represents Phase (i.e Phase I is P1 etc.). Outgoing edges from Phase nodes are possible how much to inspect Phase. For example $i : 2$ represents that Phase will be inspected twice in the Time period. Green nodes represent Game nodes, where the game for specific Phase with a specific number of inspectionsis computed. Outputs from these nodes are outgoing edges which represent probability of following nodes. Red nodes represent leaves, which have a certain value. If the testing finishes successfully with correct data then the value is 100000, otherwise ti is 0.

In conclusion, we can see that with growing number of inspections the uncertainty that pharmaceutical company observes incorrect data declines. Additionally, if the Phases are inspected optimally then Phases could not be fully covered by the inspections and it helps more effectively to divide global budget into phases and save up global budget.

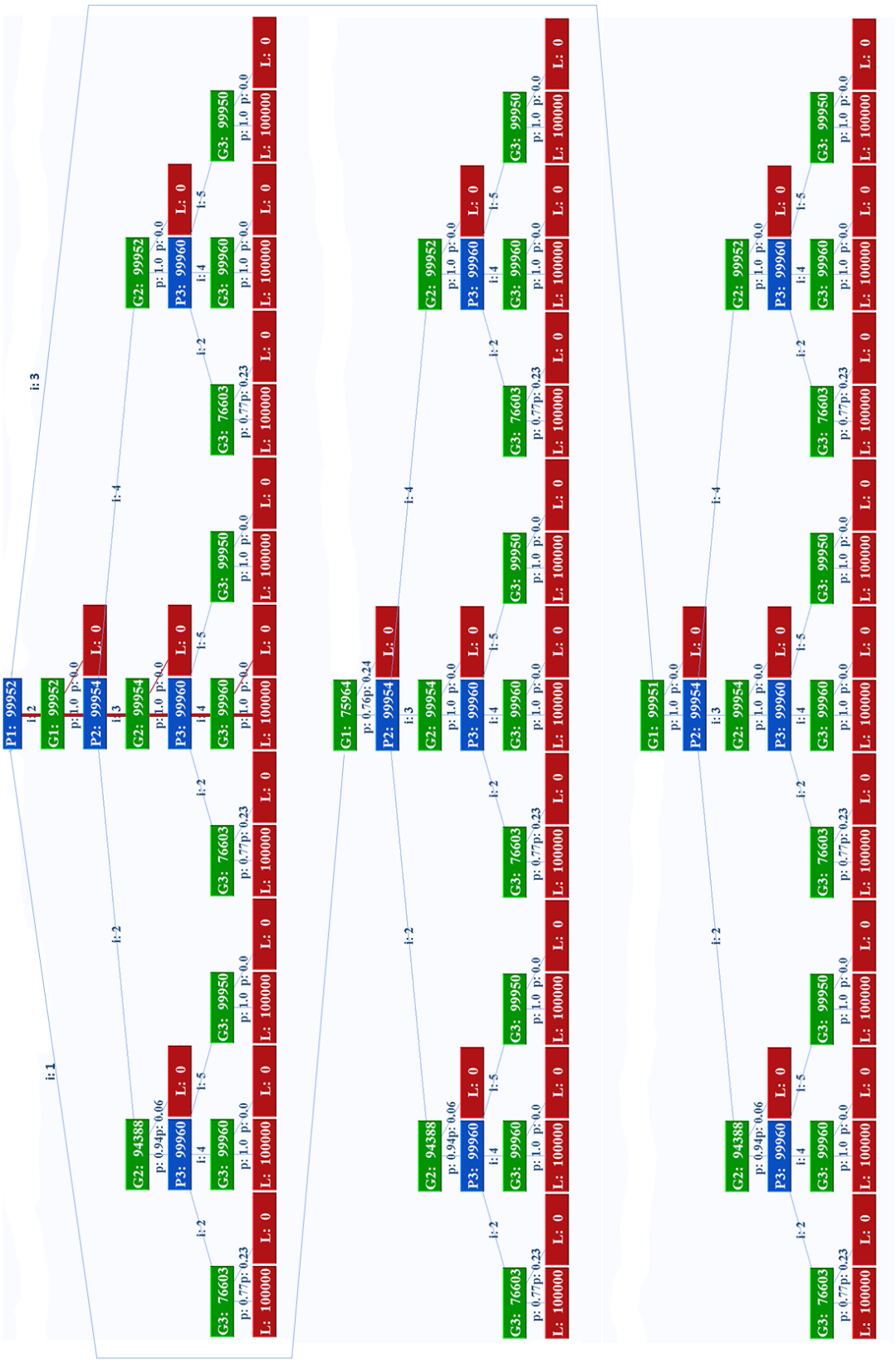


Figure 17 Budget division decision tree for budget equal to 65

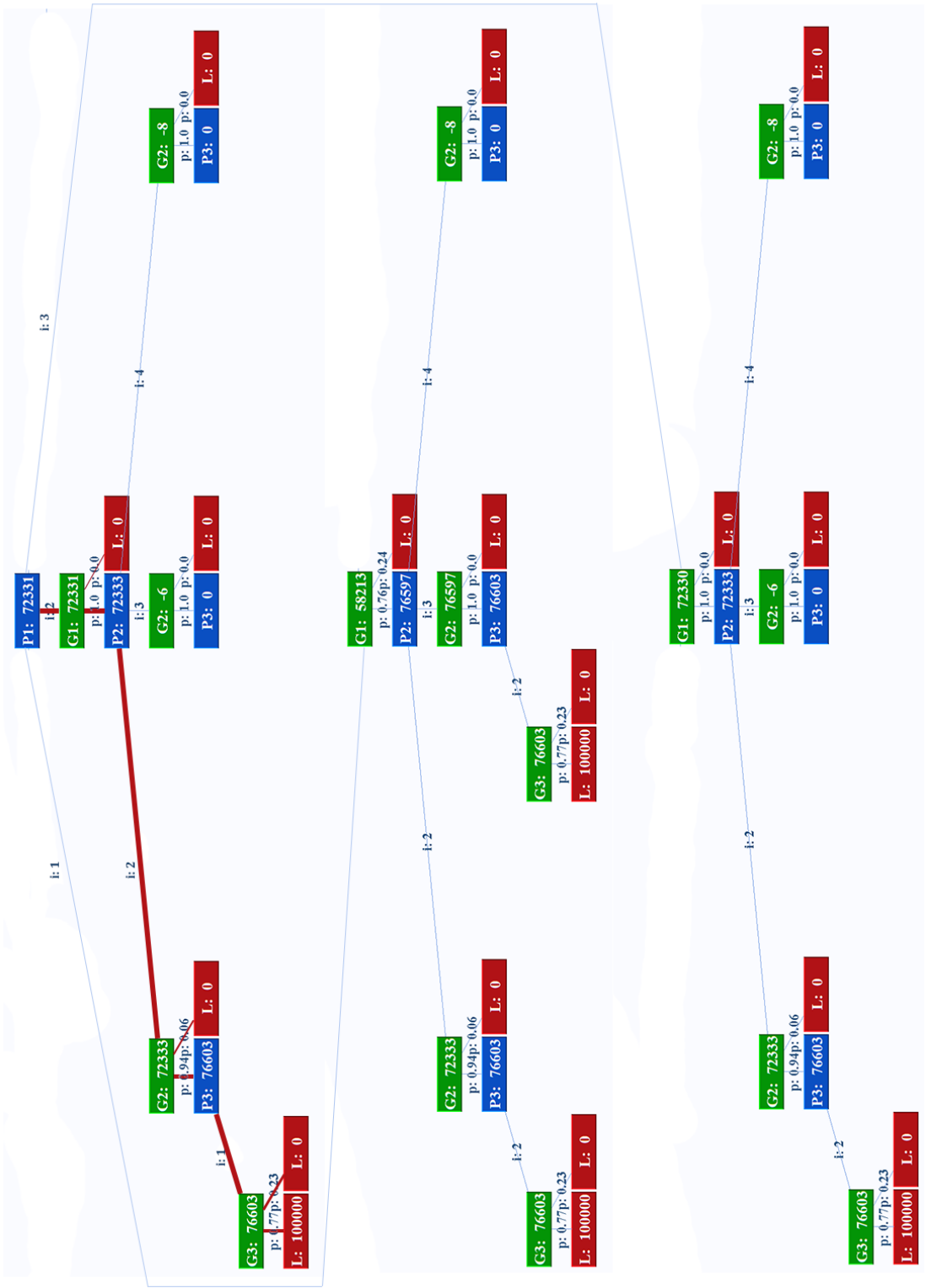


Figure 18 Budget division decision tree for budget equal to 27

8 Conclusions

The goal of this thesis was to decompose the process of clinical trials and then detect and prevent the doctor's fraud in one Phase of the clinical trials by the optimal scheduling inspections for the inspector. The area of clinical trials is very complicated. Every Phase has a number of specifications which depend on the type of the drug and on the type of Phase. Some specifications depend on the pharmaceutical company.

We have proposed a model of the inspection problem in one Phase of the clinical trial as a game between the pharmaceutical company and the doctor, who controls participants in control weeks in the Time period of Phase. The pharmaceutical company is represented be an inspector, who is paid by the pharmaceutical company and he is responsible for the correct data from each Phase.

Firstly, we formalized the game as a security zero-sum game and we searching for solution using Nash equilibrium. But the zero-sum game does not reflect differing interests of agents exactly. Thus, then we formalized the game as a Stackelbeg Security game, which is closer to reality and which is able to solve the inspection scheduling problem optimally. In this game, the inspector is leader who tries to prevent the doctor's fraud. The inspector commits to a mixed strategy which is observable for the doctor. The doctor is the follower and he is motivated to perform frauds, i.e., he wants to change the data from the testing of the Phase. This model was created for the pharmaceutical company with a limited budget for inspections, and for different types of Phases. Thus, we defined the game for a limited number of inspections and for a different number of weeks in the Time period which specifies the type of Phase of clinical trial.

As a part of formalization of Stackelberg Security game, we have created two utility functions models how the inspection scheduling problem can be represented. Firstly, we created time independent model which consider that all weeks in the Time period have the same importance for the decision about the quality of the drug. Then we extended this model and we created time dependent model which respects that different aspect of the drug can be observed in different weeks in the Time period. That implies that every week has a different importance for the decision about the quality of the drug.

The solution of the Stackelberg Security game for solving schedule problem in one Phase of clinical trial is found by the set of linear programs, where each linear program is solved by IBM CPLEX solver.

We created a scenario on which we evaluated the model of inspection scheduling problem. We found out that for a given number of possible inspections the solution computed by the SSE demotivated the doctor to perform fraud the most in comparison with greedy and uniform strategy. We tested how the doctor's incentives to perform a fraud decrease with the increased number of inspections and we compared how the strategies of the inspector and the doctor changed with different number of inspections. We demonstrated that the doctor is almost completely demotivated to perform fraud for number of inspections lower than number of control weeks in the Time period if the inspection schedule is computed using SSE. We also showed via an experiment that model of inspection problem solved by LP is hardest to solve with deployment-to-saturation ratio (Manish et. al. 2014) $(d : s) \approx 0.5$.

We fulfilled all points from the bachelor project assignment. Except the goals specified in the assignment, we have proposed a model of budget division. This model is applicable in a situation when the pharmaceutical company knows that the tested drug is effective with a high probability. Then pharmaceutical company is afraid that somebody wants to thwart clinical trial of this drug for competitive or adversarial reasons.

The model of budget division is able to plan an optimal number of inspections for each Phase of clinical trial under the constraint that the cost for all inspections in the clinical trial has to be equal or lower than the maximal global budget for the clinical trial. The model expects that one inspection in each Phase has different cost and that inspections in each Phase are scheduled with the model of inspection scheduling problem.

We evaluated this model for two types of global budgets. The firs budget was able to cover completely all Phases by the inspections and the second budget was limited. We showed that if the Phases of clinical trial were optimally inspected then the optimal number of inspections for each Phase of clinical trial is lower then number of weeks in the Time period of the Phase. Thus, even if the company has a budget to cover completely all Phases by the inspections, it is better to inspect only optimal number of days, because every extra inspection costs extra money. We showed, that it is possible to optimally divide budget even for a limited budget but we have to expect that the uncertainty that pharmaceutical company receive incorrect data is higher.

In conclusion, we described an innovative application of game theory to inspections of clinical trials. This topic allows future use and extensions. If it would be possible to extract from historical data doctor's reliability, this knowledge could be incorporated into the model to design more effective schedules. This model can be inspiration for control organizations of clinical trials as FDA which can use analogue of this inspection models to detect fraudulent behavior of pharmaceutical companies.

Bibliography

[1] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[2] Vincent Conitzer and Tuomas Sandholm. "Computing the optimal strategy to commit to". In: *Proceedings of the 7th ACM conference on Electronic commerce*. ACM. 2006, pp. 82–90.

[3] Bernhard Von Stengel and Shmuel Zamir. "Leadership with commitment to mixed strategies". In: (2004).

[4] Dmytro Korzhyk et al. "Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness." In: *J. Artif. Intell. Res.(JAIR)* 41 (2011), pp. 297–327.

[5] Hilary Paul Williams. "Model building in mathematical programming". In: (1999).

[6] Tomáš Werner. "Optimalizace". In: (2014).

[7] IBM. *Linear programming:Linear programming and CPLEX Optimizer*. [online]. [cit. 22. 4. 2015]. URL: `http://www-01.ibm.com/software/commerce/optimization/linear-programming/`.

[8] *Decision Trees: Choosing by Projecting "Expected Outcomes"*. [online]. [cit. 22. 04. 2015]. URL: `http://www.mindtools.com/dectree.html`.

[9] TreePlan. *Introduction to Decision Trees*. [online]. [cit. 11. 4. 2015]. URL: `http://www.treeplan.com/chapters/introduction-to-decision-trees.pdf`.

[10] IBM ILOG CPLEX. "12.2 user's manual". In: 2010.

[11] Michael Trick. *The simplex method*. [online]. [cit. 24. 2. 2015]. URL: `http://mat.gsia.cmu.edu/classes/QUANT/NOTES/chap7.pdf`.

[12] Nimrod Megiddo. *Pathways to the optimal set in linear programming*. Springer, 1989.

[13] Robert E Bixby. "A brief history of linear and mixed-integer programming computation". In: *Documenta Mathematica* (2012), pp. 107–121.

[14] James Pita et al. "Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*. International Foundation for Autonomous Agents and Multiagent Systems. 2008, pp. 125–132.

[15] Jason Tsai et al. "IRIS-a tool for strategic security allocation in transportation networks". In: (2009).

[16] James Pita et al. "GUARDS: game theoretic security allocation on a national scale". In: *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2011, pp. 37–44.

[17] Benjamin Ford et al. "PAWS: adaptive game-theoretic patrolling for wildlife protection". In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2014, pp. 1641–1642.

[18] Pharma. *Clinical Trials: The Phases of Drug Testing Approval*. [online]. [cit. 29. 10. 2014]. URL: `http://www.phrma.org/innovation/clinical-trials`.

[19] Merck. *About clinical trials*. [online]. [cit. 15. 11. 2014]. URL: http://www.merck.com/clinical-trials/about-clinical-trials.html.

[20] Britannica. *Vaccine*. [online]. [cit. 27. 1. 2015]. URL: http://www.britannica.com/EBchecked/topic/621274/vaccine.

[21] FDA. *Drug*. [online]. [cit. 27. 1. 2015]. URL: http://www.fda.gov/Drugs/InformationOnDrugs/ucm079436.htm.

[22] CenterWatch. *Overview of Clinical Trials*. [online]. [cit. 29. 10. 2014]. URL: http://www.centerwatch.com/clinical-trials/overview.aspx.

[23] Avik Roy. *Stifling New Cures: The True Cost of Lengthy Clinical Drug Trials*. [online]. [cit. 12. 12. 2014]. URL: http://www.manhattan-institute.org/html/fda_05.htm.

[24] Ashwaria Gupta. *Fraud and misconduct in clinical research: A concern*. [online]. [cit. 3. 2. 2015]. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3700330/.

[25] Eliot Marshall. *Cheating Rife in Clinical Trial*. [online]. [cit. 4. 2. 2015]. URL: http://news.sciencemag.org/2000/08/cheating-rife-clinical-trial.

[26] Fiona Barry. *Clinical trials: the technology catching cheating volunteers*. [online]. [cit. 4. 2. 2015]. URL: http://www.outsourcing-pharma.com/Clinical-Development/Clinical-trials-the-technology-catching-cheating-volunteersl.

[27] Office of Good Clinical Practice. *Information Sheet Guidance For IRBs, Clinical Investigators, and Sponsors FDA Inspections of Clinical Investigators*. [online]. [cit. 24. 2. 2015]. URL: http://www.fda.gov/downloads/RegulatoryInformation/Guidances/UCM126553.pdf.

[28] Zhengyu Yin et al. "Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2010, pp. 1139–1146.

[29] Manish Jain, Kevin Leyton-Brown, and Milind Tambe. "A Study of Phase Transitions in Security Games". In: *Target* 1.5 (2012), p. 5.