



# THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Scheduling Internal Audit Activities: A Stochastic Combinatorial Optimization Problem

**Citation for published version:**

Rossi, R, Tarim, SA, Hnich, B, Prestwich, S & Karacaer, S 2010, 'Scheduling Internal Audit Activities: A Stochastic Combinatorial Optimization Problem' *Journal of Combinatorial Optimization*, vol. 19, no. 3, pp. 325-346. DOI: 10.1007/s10878-009-9207-z

**Digital Object Identifier (DOI):**

[10.1007/s10878-009-9207-z](https://doi.org/10.1007/s10878-009-9207-z)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Publisher's PDF, also known as Version of record

**Published In:**

*Journal of Combinatorial Optimization*

**Publisher Rights Statement:**

Published open access under CC BY License.

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



## Scheduling internal audit activities: a stochastic combinatorial optimization problem

Roberto Rossi · S. Armagan Tarim ·  
Brahim Hnich · Steven Prestwich ·  
Semra Karacaer

Published online: 21 January 2009

© The Author(s) 2009. This article is published with open access at Springerlink.com

**Abstract** The problem of finding the optimal timing of audit activities within an organisation has been addressed by many researchers. We propose a stochastic programming formulation with Mixed Integer Linear Programming (MILP) and Constraint Programming (CP) certainty-equivalent models. In experiments neither approach dominates the other. However, the CP approach is orders of magnitude faster for large audit times, and almost as fast as the MILP approach for small audit times.

---

Roberto Rossi is supported by Science Foundation Ireland under Grant No. 03/CE3/I405 as part of the Centre for Telecommunications Value-Chain-Driven Research (CTVR) and Grant No. 05/IN/1886. S. Armagan Tarim and Brahim Hnich are supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027.

R. Rossi (✉) · S. Prestwich

Cork Constraint Computation Centre, University College, 14 Washington St. West, Cork, Ireland  
e-mail: [rrossi@4c.ucc.ie](mailto:rrossi@4c.ucc.ie)

S. Prestwich

e-mail: [s.prestwich@4c.ucc.ie](mailto:s.prestwich@4c.ucc.ie)

R. Rossi

Centre for Telecommunication Value-Chain Driven Research, University College, Dublin, Ireland  
e-mail: [robros@gmail.com](mailto:robros@gmail.com)

S.A. Tarim · S. Karacaer

Department of Management, Hacettepe University, Ankara, Turkey

S.A. Tarim

e-mail: [armagan.tarim@hacettepe.edu.tr](mailto:armagan.tarim@hacettepe.edu.tr)

S. Karacaer

e-mail: [semra@hacettepe.edu.tr](mailto:semra@hacettepe.edu.tr)

B. Hnich

Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey  
e-mail: [brahim.hnich@ieu.edu.tr](mailto:brahim.hnich@ieu.edu.tr)

This work generalises a previous approach by relaxing the assumption of instantaneous audits, and by prohibiting concurrent auditing.

**Keywords** Uncertainty · Audit scheduling · Combinatorial optimization · Mathematical programming · Constraint programming

## 1 Introduction

Based on costs and benefits that change over time, the focus of the internal audit scheduling problem is how often to conduct an internal audit on an auditable unit. Auditable units are the units upon which internal control procedures are applied, in order to safeguard assets and assure the reliability of information flows. The scope of auditable units depends on organizational characteristics: they could be organizational units (finance, accounting department), geographic regions (branches, cities) or activities (budgeting, purchasing, etc.) (Boritz and Broca 1986).

The problem of finding the optimal timing of audit activities within an organisation has been addressed by many researchers including Wilson and Ranson (1971), Hughes (1977), Boritz and Broca (1986), and Knechel and Benson (1991). The first study of audit scheduling was by Wilson and Ranson (1971) who found the audit frequency that minimizes the discounted present value of losses and audit costs. Audit costs are assumed to be incurred at a uniform rate, while losses in the absence of auditing are assumed to rise exponentially from zero to an asymptotic level. After an audit is conducted, the losses drop to zero but start to accrue until the next audit. In Hughes (1977) the audits are chosen at the beginning of each of an infinite number of periods, conditional upon available information concerning the state of internal control system, and a model is proposed for determining the optimal timing of internal audits. The model proposed by Boritz and Broca (1986) determines the optimal audit interval, assuming that expected losses accrue if a unit remains unaudited and audit cost is incurred each time the decision to audit is made. In order to use their formulation, each audit unit is assessed using an index of loss riskiness called the Audit Unit Priority Score (AUPS). The parameters of the model (i.e. the shape of the loss function and the rate of increase in losses) are determined through auditors' judgmental process, which gives auditors flexibility in the scheduling of audit activities.

In a recent paper by Tarim et al. (2008) a stochastic version of the internal audit scheduling problem is formulated under relatively relaxed assumptions. Unlike previous models that determine optimal timing for one audit unit, their model determines the optimal timing of audit activities for multiple audit units. This is important because many firms have more than one auditable unit to which audit resources must be allocated. Moreover, their formulation takes into account uncertainty in the losses accrued in the absence of auditing, and employs a chance-constraint to keep the expected losses below a certain level with a given probability. However, in Tarim et al. (2008) it is assumed that audit activities are instantaneous, i.e. that conducting an audit does not take any time. The computational issues are not addressed by Tarim et al.

Mixed Integer Linear Programming (MILP) (Nemhauser and Wolsey 1988) and Constraint Programming (CP) (Hanus 2001) are two orthogonal approaches used to

address combinatorial problems. MILP-based methods are rooted at the area of Operations Research (Nemhauser and Wolsey 1988), whereas CP-based methods are the result of research by the Artificial Intelligence community in the areas of Logic Programming and Constraint Satisfaction (Colmerauer 1985; Van Hentenryck 1989; Tsang 1993). MILP and CP methods have both been successfully applied to solve diverse problems such as network synthesis, crew scheduling, planning, and capital budgeting. Determining which type of problems or instances are best solved by which method is an active research area. This is the line of research pursued in this paper. We relax the assumption of instantaneous audits and propose a stochastic programming formulation for this important class of combinatorial problems involving uncertainty. To solve this stochastic program we develop two alternatives deterministic equivalent models: a MILP model and a CP model. The two approaches are complementary in the sense that for some instances MILP is superior and for other instances CP is superior. Our numerical experiments show that the certainty-equivalent MILP formulation is efficient when the time to perform an audit is relatively short. However, as the audit time gets longer our CP model proved to be significantly much more effective than the MILP model.

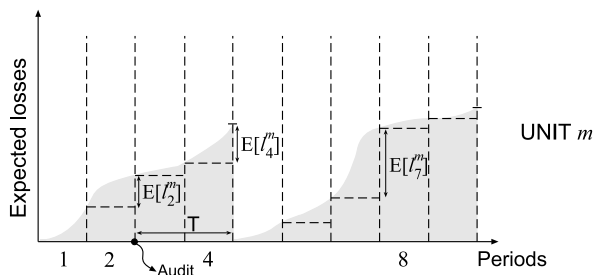
The paper is organised as follows. Section 2 describes the problem, Sect. 3 provides a stochastic programming formulation of the problem, Sect. 4 surveys possible solution methods, Sect. 5 reports experimental results comparing methods, and Sect. 6 concludes the paper. Finally, in the Appendix we provide a complete list of the notation adopted in the paper.

### 2 Problem statement

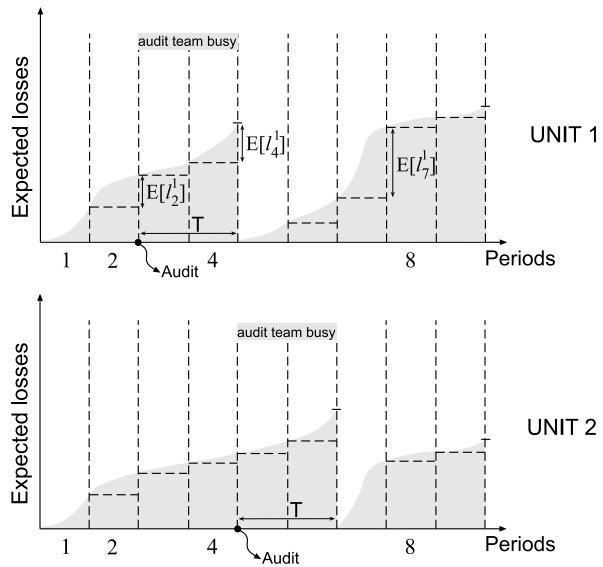
We consider a planning horizon comprising  $N$  time periods. We are given a set of  $M$  audit units over which random losses may accrue over time. In particular,  $l_t^m$  corresponds to the losses that accrue in audit unit  $m$  during period  $t$ .  $l_t^m$  is a random variable with a known probability density function  $g_{l_t^m}(l_t^m)$ . For convenience, losses in each period are assumed to be normally distributed with a constant coefficient of variation:  $\rho = \sigma_t^m / \mu_t^m$  in this problem, but this assumption may be relaxed without loss of generality. The distribution of losses may vary from period to period, i.e. it is non-stationary. Losses in different time periods are assumed to be independent. Figure 1 illustrates expected losses on a single auditable unit.

Without loss of generality, we consider the case in which a single audit team has to be employed to keep losses under control. Auditing is a time-consuming task, and we

**Fig. 1** Expected losses;  $E[l_t^m]$  denotes the expected value of  $l_t^m$



**Fig. 2** Multiple units



assume that the team is given a strict deadline for performing an audit. Specifically, an audit must be completed in  $T$  time periods ( $T > 0$ ). Therefore after  $T$  periods the accrued losses will drop to zero. If a team has already started auditing a unit at a given period, then no other audit can be initiated during this period for the given audit team. Figure 2 depicts such a situation in which an audit duration of 2 periods is assumed. An audit scheduled for unit 1 at the beginning of period 3 rules out any following audit for unit 2 until period 5.

Note that the timing of audits are fixed once and for all at the beginning of the planning horizon, and cannot be changed thereafter even if it is suspected that certain auditable units have accrued unexpected losses. The objective is to find the optimal audit schedule while respecting the maximum loss level criterion. That is, the invariant audit costs (i.e. fixed audit costs incurred each time an audit is conducted) and expected total discounted audit losses (i.e. cumulative losses accrued at the end of each period) are minimized by satisfying a maximum loss level constraint, which in this problem is defined by specifying a minimum probability  $\alpha$  that the losses will not exceed a predetermined level  $\bar{L}$  (allowed maximum loss) in any given audit period for any auditable unit.

*Example 1* In what follows we will employ a running example to better exemplify the above concepts. We consider the following simple instance:

- $M$ : the total number of audit units, equal to 2
- $N$ : the number of periods in the planning horizon, equal to 6
- $T$ : the duration of an audit in time periods, equal to 2
- $a$ : the fixed cost incurred each time an audit is conducted, equal to 100
- $h$ : the loss discount factor measuring the opportunity cost associated to a given loss level, equal to 1

$\bar{L}$ : a threshold indicating the maximum allowed loss level in each period, equal to 200

$\alpha$ : the probability of not exceeding the loss threshold  $\bar{L}$ , equal to 0.95.

We assume the losses accrued in each period to be normally distributed with a constant coefficient of variation  $\rho = 0.2$ , where  $\rho = \sigma_t^m / \mu_t^m$ . The expected value,  $\mu_t^m$ , for the losses in each period  $t$  and for each audit unit  $m$  is respectively  $\{50, 30, 50, 30, 50, 30\}$  in each period  $t = 1, \dots, 6$  for audit unit 1, and  $\{10, 20, 30, 40, 50, 60\}$  in each period for audit unit 2.

### 3 Stochastic programming formulation

*Stochastic programming* (Birge and Louveaux 1997) is a well known modeling technique that deals with problems where uncertainty comes into play. Problems of optimization under uncertainty are characterized by the necessity of making decisions without knowing what their full effect will be. Stochastic programming needs to represent uncertain elements of the problem. Typically random variables are employed to model this uncertainty to which probability theory (Ventsel 1979; Jeffreys 1961) can be applied. For this purpose such uncertain elements must have a known probability distribution. The typical requirement in stochastic programs is to maintain certain constraints, called *chance constraints* (Charnes and Cooper 1959), satisfied at a prescribed level of probability. The objective is typically related to the minimization/maximization of some expectation on the problem costs.

The stochastic programming model we propose balances the discounted cost of losses accrued due to lack of audits with the cost of conducting audits. Taking the discounted cost of losses into account is particularly relevant when the cost of money must be considered. Consider, for instance, the situation in which some losses are due to a specific reason in a given period. Then in the following periods this loss will affect company assets until the originating factor is discovered by an audit and cleared. Obviously these effects will have a higher impact the longer it takes to clear such an originating factor. Consider, for instance, the case in which a company's tax liabilities are overestimated. The capital tied into tax liabilities could be invested in a more profitable way if the accounts were not flawed. In this case the discount factor would reflect the opportunity cost associated with the fact that capitals may be invested in a more profitable way if an audit were scheduled.

We employ the expected value criterion to minimize the sum of the expected discounted period losses and audit costs over an  $N$  period planning horizon. Let us consider, without loss of generality, an initial loss levels  $L_1^m$  set to any non-negative values for each audit unit  $m = 1, \dots, M$ . Let

$L_t^m$ : the loss level in audit unit  $m$  at the beginning of period  $t$ .

The objective function below and the following constraints give the optimum audit timing for each audit unit by minimizing  $E[TC]$ , that is the sum of expected audit costs and discounted period losses that are expected to accrue in the absence of

auditing.

$$\begin{aligned} \min E[TC] = & \sum_{m=1}^M \int_{l_1^m} \int_{l_2^m} \dots \int_{l_N^m} \sum_{t=1}^N (aK_t^m + h(L_t^m + l_t^m)) \\ & \times g_{l_1^m}(l_1^m) g_{l_2^m}(l_2^m) \dots g_{l_N^m}(l_N^m) d(l_1^m) d(l_2^m) \dots d(l_N^m) \end{aligned} \quad (1)$$

where

- $M$ : the total number of audit units
- $N$ : the number of periods in the planning horizon
- $a$ : the amount of cost incurred each time an audit is conducted
- $h$ : the loss discount factor measuring the opportunity cost associated to a given loss level
- $K_t^m$ : a variable that takes the value of 1 if an internal audit (lasting  $T$  periods) is started for audit unit  $m$  in period  $t$ , otherwise 0.

The above objective function is subject to several constraints. If ( $K_t^m = 1$ ) an internal audit is conducted at the beginning of period  $t$  (i.e. at the end of period  $t - 1$ ), then the loss level at the beginning of period  $t + T$  should be 0. Yet, if an internal audit is not conducted, the loss level at the beginning of period  $t + T$  will be equal to the loss level at the beginning of the preceding period plus the loss accrued during the preceding period. This can be expressed as

$$L_{t+T}^m \geq L_{t+T-1}^m + l_{t+T-1}^m - \mathfrak{M}K_t^m, \quad m = 1, \dots, M, \quad t = 1, \dots, N - T, \quad (2)$$

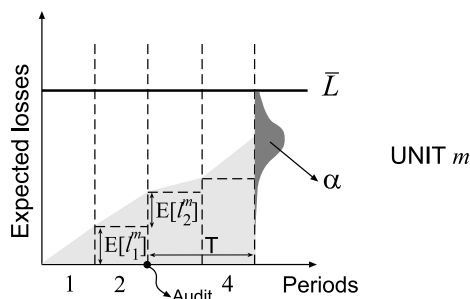
where  $\mathfrak{M}$  is some very large number. Obviously in the first  $T$  periods no audit can be completed, therefore

$$L_t^m \geq L_{t-1}^m + l_{t-1}^m, \quad m = 1, \dots, M, \quad t = 1, \dots, T. \quad (3)$$

For convenience we consider  $l_t^m = L_t^m = 0$ , for  $\{t \mid t < 1\}$ .

Now consider a plan for audit unit  $m$ , which schedules  $r$  audits over the  $N$  period planning horizon with audits conducted at  $\{A_1^m, \dots, A_r^m\}$ , where  $A_j^m > A_{j-1}^m$ ,  $A_r^m \leq N - T$ . For convenience  $A_1^m = 1 - T$ , because the initial loss level is set to 0;  $A_{r+1}^m = N - T + 1$  is defined as the earliest period for which an associated audit

**Fig. 3** Chance-constraint on the maximum loss level. Assuming losses to be normally distributed:  $\alpha$  is the desired minimum probability (area marked in the figure) that the loss level in any time period will not exceed a subjectively determined level,  $\bar{L}$



would be completed only after the end of the horizon. The associated audits will take place at the beginning of periods  $A_i^m, i = 1$  to  $r$ . In the considered plan there are clearly no audits scheduled for audit unit  $m$  except at periods  $A_1^m, \dots, A_r^m$ . The accumulated loss level  $L_{t+1}^m$  carried over from period  $t$  to period  $t + 1$  is the loss to date since the last completed audit. This can be written as

$$L_{t+1}^m = \sum_{k=A_i^m+T}^t l_k^m, \quad A_i^m + T \leq t < A_{i+1}^m + T, \quad i = 1, \dots, r. \tag{4}$$

As defined above,  $\alpha$  is the desired minimum probability that the loss level in any time period will not exceed a subjectively determined level,  $\bar{L}$  (Fig. 3). In this regard the chance constraint becomes

$$\Pr\{L_t^m + l_t^m \leq \bar{L}\} \geq \alpha, \quad t = 1 + T, \dots, N. \tag{5}$$

Using (4), this can be written alternatively as, for  $t \geq 1 + T$

$$\Pr\left\{\sum_{k=A_i^m+T}^t l_k^m \leq \bar{L}\right\} \geq \alpha, \quad A_i^m + T \leq t < A_{i+1}^m + T, \quad i = 1, \dots, r, \tag{6}$$

which implies

$$G_{l_{A_i^m+T}^m+l_{A_{i+1}^m+T+1}^m+\dots+l_t^m}(\bar{L}) \geq \alpha, \quad A_i^m + T \leq t < A_{i+1}^m + T, \tag{7}$$

where  $G_{l_t^m}(x) = \int_{-\infty}^x g_{l_t^m}(\tau) d\tau$  is the cumulative distribution function of  $l_t^m$ . By assuming  $G_{l_t^m}(x)$  to be strictly increasing, thus invertible, (7) can then be rewritten as

$$\bar{L} \geq G_{l_{A_i^m+T}^m+l_{A_{i+1}^m+T+1}^m+\dots+l_t^m}^{-1}(\alpha), \quad A_i^m + T \leq t < A_{i+1}^m + T, \tag{8}$$

where  $G_{l_t^m}^{-1}(\alpha)$  is the inverse cumulative distribution function (or  $\alpha$ -quantile) of  $l_t^m$ .

Since the problem has a finite planning horizon of  $N$  periods, for all the relevant cases the right-hand side of (8),  $G_{l_{A_i^m+T}^m+l_{A_{i+1}^m+T+1}^m+\dots+l_t^m}^{-1}(\alpha)$ , can be computed or possibly read from a table, once the form of  $g_{l_t^m}(\cdot)$  is decided. If the binary variable  $P_{t,j}^m$  is defined as taking a value of 1 if the most recent audit prior to period  $t$  was in period  $j$  and zero elsewhere for a given audit unit  $m$ , then (8) can be written as

$$\bar{L} \geq \sum_{j=1}^t (G_{l_j^m+l_{j+1}^m+\dots+l_t^m}^{-1}(\alpha) P_{t,j-T}^m). \tag{9}$$

There can be at most only one most recent audit prior to period  $t$ . Thus  $P_{t,j}^m$  must satisfy

$$\sum_{j=1-T}^t P_{t,j}^m = 1, \quad m = 1, \dots, M, \quad t = 1, \dots, N. \tag{10}$$



Note that losses in the first  $T$  periods cannot be controlled with respect to the threshold  $\bar{L}$  and the probability  $\alpha$ . In fact benefits from the first possible audit appear only in period  $T + 1$ . Therefore we assume, for all  $m = 1, \dots, M$ ,  $K_{1-T}^m = 1$  (according to the fact that initial losses should be equal to 0) and  $K_{i-T}^m = 0, i = 2, \dots, T$ .

The following equation (11) is necessary to identify uniquely the period in which the most recent audit prior to any period  $t$  took place. For each audit unit  $m = 1, \dots, M$

$$P_{t,j}^m \geq K_j^m - \sum_{k=j+1}^{t-T} K_k^m, \quad t = 1, \dots, N, \quad j = 1 - T, \dots, t - T. \quad (11)$$

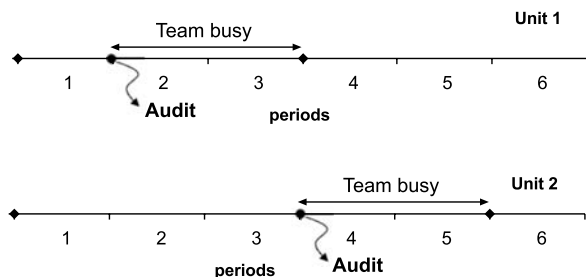
It is not common practice for internal audit teams to conduct multiple audits simultaneously. To have a modicum of resemblance to reality, as already stated, here it is assumed that a team can conduct an internal audit only for one audit unit at a given time period. In our model we shall consider the following capacity constraint

$$\sum_{k=1}^m K_t^m \leq C, \quad m = 1, \dots, M, \quad t = 1, \dots, N - T, \quad (12)$$

which states that the firm can assign at most  $C$  audit teams to conduct audits in any given time period. For simplicity, in what follows we will assume  $C = 1$ .

*Example 2* For the running example introduced in Sect. 2, in Table 1 we show the values of  $G_{l_j^m+l_{j+1}^m+\dots+l_t^m}^{-1}(\alpha)$  for audit unit  $m = 1, 2$ . Values in *italic* are those that satisfy constraint (9), that is values that stay below the loss threshold  $\bar{L} = 200$ . Underlined values identify audit cycles<sup>1</sup> in the optimal solution. The optimal audit plan is also presented graphically in Fig. 4. In this plan, in order to keep accrued losses under control, one single audit for unit 1 is scheduled at the beginning of period 2, and this audit terminates at the end of period 3. Similarly, for unit 2 a single audit is scheduled at period 4 and this audit terminates at the end of period 5. The expected total cost for this plan is 1090.

**Fig. 4** Optimal audit plan for the numerical example



<sup>1</sup>An audit cycle is a set of periods  $\{j, \dots, t\}$ ,  $j \leq t$ , where  $j$  is the first period after the completion of an audit, and no other audit is completed by period  $t$ .

**Table 1** Values of  $G_{l_j^m+l_{j+1}^m+\dots+l_t^m}^{-1}(\alpha)$  for audit unit  $m = 1, 2$ . Values in italic are those that satisfy constraint (9). That is, values that stay below the loss threshold  $\bar{L} = 200$ . Underlined values identify audit cycles in the optimal solution

<i>j</i>	<i>t</i>					
	1	2	3	4	5	6
Unit 1						
1	66.5	99.2	<u>155.3</u>	187.1	241.7	273.2
2		39.9	99.2	131.6	187.1	218.9
3			66.5	99.2	155.3	187.1
4				39.9	99.2	<u>131.6</u>
5					66.5	99.2
6						39.9
Unit 2						
1	13.3	37.4	72.3	118.0	<u>174.4</u>	241.4
2		26.6	61.9	107.7	164.2	231.2
3			39.9	86.5	143.3	210.5
4				53.2	111.1	178.9
5					66.5	135.7
6						<u>79.7</u>

We now show in detail how the values in Table 1 are computed. Let  $G_N$  be the standard normal distribution function. This function is strictly increasing, therefore  $G_N^{-1}$  is uniquely defined.<sup>2</sup>  $G_N^{-1}(0.95) = 1.645$  corresponds to the 0.95-quantile. Therefore, since all the random variables  $l_t^m, t = 1, \dots, N, m = 1, \dots, M$ , are independent and normally distributed,  $G_{l_2^1+l_3^1+l_4^1}^{-1}(0.95) = 1.645 \cdot 0.2 \cdot \sqrt{30^2 + 50^2 + 30^2} = 131.6$ . This value can be found in the first matrix presented in Table 1 at position (2, 4). It corresponds to a sequence of periods starting in period 2 and ending in period 4, where no audit is completed and where the last audit performed has been completed by the end of period 1 (therefore losses at the beginning of period 2 are null). Since  $131.6 < 200$  it follows that this set of periods constitutes a feasible audit cycle.

### 4 Solution methods

In this section we present two alternative certainty equivalent (Birge and Louveaux 1997) models for the stochastic programming model presented in the former section: a Mixed Integer Linear Programming model and a Constraint Programming model.

#### 4.1 A certainty equivalent MILP model

In Linear Programming (Dantzig 1963; Chvatal 1983; Schrijver 1986) a model, called a “program”, consists of continuous variables and linear constraints (inequalities or

<sup>2</sup>Tables are available for obtaining values of the inverse normal cumulative distribution function (also known as  $\alpha$ -quantile, Ventsel 1979).

equalities), and the aim is to optimize (minimize or maximize) a linear cost function. In matrix notation the standard form of a linear program is  $\min\{c^T x \mid Ax = b, x \geq 0\}$ , where  $c \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ . Here  $c$  represents the cost vector and  $x$  is the vector of variables. Linear programs are usually solved by the *simplex method* (Dantzig 1951) which is very fast in practice, though it has exponential worst-case time complexity. A Mixed Integer Linear Program (MILP) is a Linear Program (LP) plus integrality requirements on some decision variables. Discrete variables in MILP are often 0–1 variables representing yes/no, on/off or true/false choices. Logical conditions between 0–1 variables such as  $\vee$  (or),  $\wedge$  (and),  $\neg$  (not),  $\Rightarrow$  (if ... then), and  $\Leftrightarrow$  (if and only if) can be imposed using appropriate linear constraints (Williams 1994). Solution methods for MILP typically employ tree search in which internal nodes correspond to partial solutions, branches are choices partitioning the search space, and leaf nodes are solutions. Branching is intertwined with a relaxation to eliminate the exploration of nodes for which the relaxation is either infeasible or worse than the best solution found so far. Each node represents a partial assignment of the discrete variables, and at each node a relaxation is formed by turning the integrality requirements into bounds, thus transforming the subproblem into an LP. This LP is solved, and if the solution is not suboptimal then descendant nodes are formed by branching on the fractional relaxation value of a discrete variable. The historical popularity of MILP derives from Dantzig’s discovery that the vocabulary of LP is surprisingly versatile in many applications. MILP-based methods have been developed over the last four decades by the Operations Research community (Nemhauser and Wolsey 1988).

The mathematical programming model of the previous section, as presented, is a stochastic nonlinear combinatorial optimization model, which is extremely complex to solve. In this section we adopt the static-dynamic uncertainty strategy, proposed by Bookbinder and Tan (1988) to solve their stochastic inventory lot-sizing problem, and apply it to the mathematical programming model of Sect. 3. The model can be expressed as minimizing the objective function given in (1) subject to the constraints (2–4), (9–12), and non-negativity and 0/1 integrality conditions for  $L_t^m$ ,  $K_t^m$  and  $P_{i,j}^m$ .

In our internal audit scheduling problem the analysis is completed at the beginning of the planning horizon by taking expectations (see Bookbinder and Tan 1988). Hence the deterministic equivalent model for the original chance-constrained stochastic programming model is obtained. The resultant model is in the form of a mixed-integer program, given below, in which the expected value operator is denoted by  $E[\cdot]$ .

$$\min \sum_{m=1}^M \left( \sum_{t=1}^N a K_t^m + \sum_{t=1}^N h E[L_t^m + l_t^m] \right) \tag{13}$$

$$\text{subject to, for } m = 1, \dots, M, \tag{14}$$

$$E[L_1^m] = 0, \tag{15}$$

$$E[L_{t+1}^m] \geq E[L_t^m] + E[l_t^m], \quad t = 1, \dots, T, \tag{16}$$

$$E[L_{t+T}^m] \geq E[L_{t-1+T}^m] + E[l_{t-1+T}^m] - \mathfrak{M} K_t^m, \quad t = 1, \dots, N - T, \tag{17}$$

$$\sum_{k=1}^M \sum_{h=1}^{\min(t+T-1,N)} K_h^k \leq 1, \quad t = 1, \dots, N, \tag{18}$$

$$\bar{L} \geq \sum_{j=1}^t G_{l_j^m+l_{j+1}^m+\dots+l_t^m}^{-1}(\alpha) \cdot P_{t,j-T}^m, \quad t = 1, \dots, N, \tag{19}$$

$$\sum_{j=1}^t P_{t,j}^m = 1, \quad t = 1 - T, \dots, N, \tag{20}$$

$$P_{m,t,j} \geq K_j^m - \sum_{k=j+1}^{t-T} K_k^m, \quad j = 1 - T, \dots, t - T, \tag{21}$$

$$K_{1-T}^m = 1, \tag{22}$$

$$K_{t-T}^m = 0, \quad t = 2, \dots, T, \tag{23}$$

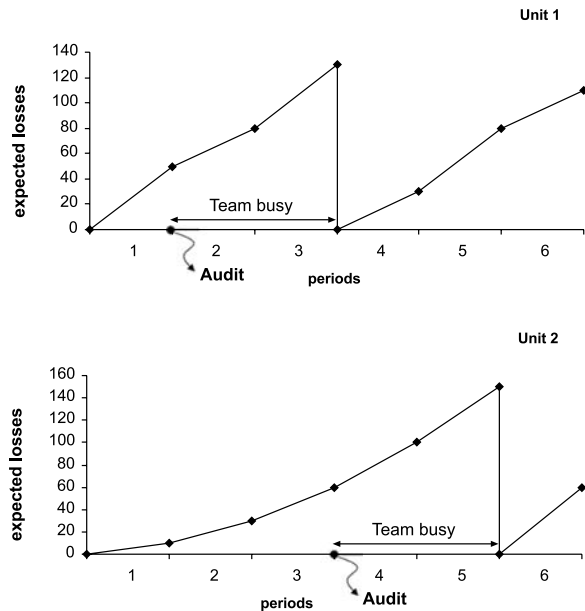
$$E[L_t^m] \geq 0, \tag{24}$$

$$K_t^m, P_{t,j}^m \in \{0, 1\}, \quad t = 1, \dots, N, \quad j = 1 - T, \dots, t. \tag{25}$$

This model thus determines the optimal audit schedule by balancing the fixed audit costs and discounted expected period losses that accrue in the absence of auditing. The problem is to determine the values of the 0/1 integer variables,  $K_t^m$  for  $m = 1, \dots, M, t = 1, \dots, N$ , and  $P_{t,j}^m$  for  $t = 1, \dots, N, j = 1 - T, \dots, t$ , and the non-negative continuous variable  $E[L_t^m]$  for  $t = 1, \dots, N$ , that minimize the objective function. The times of the audit reviews in each audit unit  $m$  are given by the values of  $t$  such that  $K_t^m = 1$ . Constraint (15) states that the initial losses are equal to 0. Constraint (16) lets expected losses accumulate in the first  $T$  periods for each audit unit, since no audit can be terminated before period  $T + 1$ . Constraint (17) states that if an audit is planned in period  $t$ , then expected losses must drop to zero in period  $t + T$ , as soon as the audit terminates, while if no audit is planned in period  $t$ , then expected losses in period  $t + T$  must be equal to the expected losses accumulate till the beginning of the previous period ( $t + T - 1$ ) plus the expected losses accrued in such a period. Constraint (18) prevents multiple audits in any given period. If an audit team starts an audit in period  $t$  on a given unit, this means that no other audit can be performed on any unit before period  $t + T$ . Constraints (19–21) implement (8) and therefore they identify feasible audit schedules, that is those for which losses never exceed the given threshold  $\bar{L}$  more than  $\alpha$  percent of the times.

*Example 3* By employing the mathematical programming model presented in this section we can solve the running example originally presented in Sect. 2. The optimal plan, which we already described in Sect. 3, is shown in Fig. 5. In this picture, we also show the expected losses accumulated in each period and computed by the MILP model. The plan schedules one single audit for unit 1 at period 2, and for unit 2 at period 4. The expected total cost for this plan is 1090. Note that for each audit, losses drop to 0 only after  $T = 2$  periods, which is in fact the time required to perform an audit. It is also clear from the plan shown that the audit team cannot perform multiple audits at any given time period.

**Fig. 5** Optimal audit plan for the numerical example



## 4.2 A certainty equivalent CP model

A *Constraint Satisfaction Problem* (CSP) (Apt 2003; Brailsford et al. 1999; Lustig and Puget 2001) is a triple  $\langle V, C, D \rangle$  where  $V$  is a set of decision variables,  $D$  is a function mapping each element of  $V$  to a domain of potential values, and  $C$  is a set of constraints stating allowed combinations of values for subsets of variables in  $V$ . A *solution* to a CSP is simply a set of values of the variables such that the values are in the domains of the variables and all the constraints are satisfied. We may also be interested in finding a feasible solution that minimizes or maximizes the value of a given objective function over a subset of the variables.

We now recall some key concepts in *Constraint Programming* (CP): constraint filtering algorithm, constraint propagation and arc-consistency (Regin 2003). A *filtering algorithm* is typically associated with a constraint, and removes values from the domains of its variables that cannot belong to any solution of the CSP. These algorithms are repeatedly called until no new deduction can be made, a process called *propagation*. In conjunction with this process CP uses a search procedure (typically a backtracking algorithm) in which filtering is systematically applied whenever the domain of a variable is modified. One of the most interesting properties of a filtering algorithm is *arc-consistency*: we say that a filtering algorithm associated with a constraint establishes arc-consistency if it removes all the values from the domains of the variables involved in the constraint that are not consistent with the constraint. Studies on arc-consistency are often limited to binary constraints, but modeling problems by means of binary constraints has drawbacks: they are not very expressive, and their domain reduction is typically weak. To overcome both these drawbacks, constraints that capture a relation among a non-fixed number of variables were introduced. These constraints are not only more expressive than the equivalent aggregation

of simple constraints, but they can be associated with more powerful filtering algorithms that take into account the simultaneous presence of several simple constraints to further reduce the domains of the variables. These more powerful constraints are called *global constraints*. One of the best-known examples is the `alldiff` constraint (Regin 1994), both because of its expressiveness and its efficiency in establishing arc-consistency. A comprehensive and up-to-date survey of the state of knowledge regarding CP is Apt (2003), while a general account of the CP–MP relationship is given by Brailsford et al. (1999) and Lustig and Puget (2001).

In this section we propose a CP reformulation for the mathematical programming model in Sect. 4.1. This reformulation follows the approach originally proposed in Tarim and Smith (2008), it exploits non-binary (global) constraints and other features of CP. The CP model, similarly to the mathematical programming one, can be expressed as minimizing the objective function given in (1). But, as we shall see, in the CP model constraints (2–4) and (9–12) are now reformulated and expressed in a more compact and readable way. Furthermore, as we will see, the number of decision variables employed is dramatically reduced as we do not employ anymore the binary decision variables  $P_{i,j}^m$ . The number of constraints is also significantly reduced. The CP model is as follows.

$$\min \sum_{m=1}^M \left( \sum_{t=1}^N aK_t^m + \sum_{t=1}^N hE[L_t^m + l_t^m] \right) \tag{26}$$

subject to, for  $m = 1, \dots, M$ , (27)

$$E[L_1^m] = 0, \tag{28}$$

$$E[L_{t+1}^m] \geq E[L_t^m] + E[l_t^m], \quad t = 1, \dots, T, \tag{29}$$

$$K_t^m = 1 \rightarrow E[L_{t+T}^m] = 0, \quad t = 1, \dots, N - T, \tag{30}$$

$$K_t^m = 0 \rightarrow E[L_{t+T}^m] = E[L_{t-1+T}^m] + E[l_{t-1+T}^m], \quad t = 1, \dots, N - T, \tag{31}$$

$$\sum_{k=1}^M \sum_{h=1}^{\min(t+T-1, N)} K_h^k \leq 1, \quad t = 1, \dots, N, \tag{32}$$

$$\Phi \left[ m, t + T, \max \left( 1, \max_{j=1, \dots, t} (j + T) \cdot K_j^m \right) \right] \geq 0, \quad t = 1 - T, \dots, N - T, \tag{33}$$

where

$$\Phi[m, t, j] = \bar{L} - G_{l_j^m + l_{j+1}^m + \dots + l_t^m}^{-1}(\alpha), \quad t = 1, \dots, N, \quad j = 1, \dots, N.$$

The objective function, as in the mathematical programming model, balances the fixed audit costs and discounted expected period losses that accrue in the absence of auditing. Constraint (28) states that the initial losses are equal to 0. Constraint (29) lets losses accumulate in the first  $T$  periods for each audit unit, since no audit can be terminated before period  $T + 1$ . Constraint (30) states that if an audit is planned in

period  $t$ , then losses must drop to zero in period  $t + T$ , as soon as the audit terminates. Conversely, constraint (31) states that if no audit is planned in period  $t$ , then losses in period  $t + T$  must be equal to the losses accumulated till the beginning of the previous period ( $t + T - 1$ ) plus the losses accrued in such a period. These two non-linear constraints are equivalent to constraint (17) in the mathematical programming formulation. Constraint (32), similarly to constraint (18), prevents multiple audits in any given period. If an audit team starts an audit in period  $t$  on a given unit, this means that no other audit can be performed on any unit before period  $t + T$ . Constraint (33) identifies feasible audit schedules, that is those for which losses never exceed the given threshold  $\bar{L}$  more than  $\alpha$  percent of the time. This constraint replaces the set of constraints (19–21). The model given in (26–33) can be directly implemented using the OPL optimization programming language (Van Hentenryck 1999). It should be noted that, in OPL, constraint (33) is implemented using the `element(I, A, J)` constraint (Van Hentenryck and Carillon 1988). The `element` constraint holds iff  $A[I] = J$ , where  $I$  and  $J$  are decision variables, and  $A$  is an array of decision variables.

## 5 Experiments

In this section, we compare the computational performance of the MILP formulation, presented in Sect. 4.1, versus the CP equivalent formulation, presented in Sect. 4.2, on a number of test problems.

Computational tests are performed on a 1.5 GHz, 2 GB RAM, Centrino machine using ILOG Cplex 9.0 (Ilog 2007a) in OPL Studio 3.7 (Ilog 2007b). The packages are used with their default settings.

In the MILP model, the  $\mathfrak{M}$  in constraint (17) must have a numerical value. It is well known that the computational performance of the MILP model can be improved by choosing  $\mathfrak{M}$  as small as possible, without ruling out any possible solution. It is also clear that in different time periods, the corresponding  $\mathfrak{M}$  may have been assigned different numerical values. One way of generating such  $\mathfrak{M}$  is by observing that, by assuming a reasonably high service level (that is  $\alpha > 0.5$ ) the loss level will never exceed  $\bar{L}$ . Hence,  $\mathfrak{M} = \bar{L}$ .

It should be also noted that, while the integer program is treated in its matrix form, and different heuristics are used to choose the variable to branch on based on the solution of the LP relaxation that is solved at each node, in a CP approach the user specifies the branching strategy in terms of the formulation of the problem. The following search strategy is employed in solving the CP model proposed:  $K_t^m = 0$  and 1 are tried in order, for all  $m \in \{1, \dots, M\}$ , for all  $t \in \{1, \dots, N\}$ .

### 5.1 Experimental settings

The design of the test problems is as follows. We consider the following inputs:

$M$ : the total number of audit units, equal to 5

$N$ : the number of periods in the planning horizon, taking values in {20, 30, 40}

$T$ : the duration of an audit in time periods, taking values in {1, ..., 6}

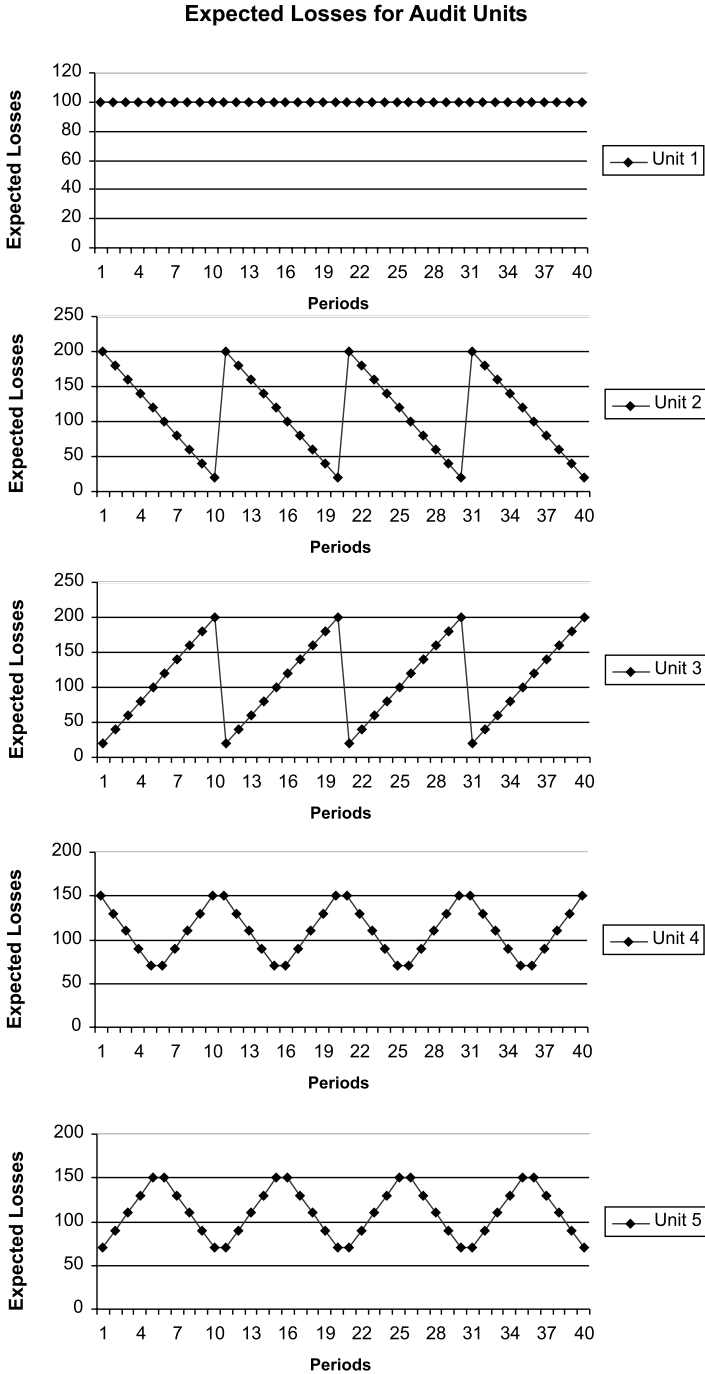


Fig. 6 Expected value  $\mu_t^m$  for the losses in each period  $t$  and for each audit unit  $m$



- $a$ : the amount of cost incurred each time an audit is conducted, taking values in  $\{500, 750, 1000\}$
- $h$ : the loss discount factor measuring the opportunity cost associated to a given loss level, equal to 1
- $\bar{L}$ : a threshold indicating the maximum allowed loss level in each period, taking values in  $\{1500, 2500, 3500\}$
- $\alpha$ : the probability of not exceeding the loss threshold  $\bar{L}$ , equal to 0.95.

We assume the losses accrued in each period to be normally distributed with a constant coefficient of variation  $\rho \in \{0.15, 0.3\}$ , where  $\rho = \sigma_t^m / \mu_t^m$ . The expected value  $\mu_t^m$  for the losses in each period  $t$  and for each audit unit  $m$  is shown in Fig. 6.

The total number of test problems generated is 108. We further partition our set of problem instances into two classes as follows:

- The instances where the audit time is 1, i.e.,  $T = 1$  (18 instances).
- The instances where the audit time is greater than 1, i.e.,  $T \in \{2, \dots, 6\}$  (90 instances).

We now analyze each set separately.

#### 5.1.1 Instances for which $T = 1$

For each test problem the solution time (in seconds), for both the MILP and the CP approach, is given in Table 2. In this table italic figures highlight the approach that produced the best run time. In Table 3 instead we reported for the MILP approach and for the CP approach, respectively, the simplex iterations performed and the nodes explored. In this first set of 18 instances, where  $T = 1$ , the MILP approach always dominates the CP approach in terms of run time. Nevertheless the discrepancy between the two approaches reaches only one order of magnitude in the worst case. In the average case MILP is faster than CP by a factor of 7.7.

#### 5.1.2 Instances for which $T > 1$

For each test problem the solution time (in seconds), for both the MILP and the CP approach, is given in Table 4. In this table, again italic figures highlight the approach that produced the best run time, while those instances for which the figures are underlined are infeasible. In Table 5 we reported for the MILP approach and for the CP approach, respectively, the simplex iterations performed and the nodes explored.

In contrast to what we observed in the first set of 18 instances, where  $T = 1$ , in this second set of 90 instances, where  $T > 1$ , the CP approach always dominates the MILP approach in terms of run time. When  $T > 1$  the MILP approach does not scale particularly well with respect to  $N$ ,  $\bar{L}$  and  $\rho$ . Instances with a large  $N$ ,  $T$ ,  $\bar{L}$  and  $\rho$  require, in fact, up to more than 30000 seconds to be solved. For these instances CP is able to quickly prove optimality or efficiently detect infeasibility. In contrast, CPLEX requires several simplex iterations and a long time to prove infeasibility. The discrepancy between the two approaches for infeasible problems reaches a factor of

**Table 2** Computational times (in sec) for the MILP approach (MILP) and for the CP approach (CP). Italic figures in the table highlight the approach that produced the best run time

$\bar{L}$	1500			2500			3500			$T$	$N$
	$a$	500	750	1000	500	750	1000	500	750		
$\rho = 0.15$											
MILP	<i>1.1</i>	<i>0.98</i>	<i>0.7</i>	<i>1.4</i>	<i>1.5</i>	<i>0.56</i>	<i>0.72</i>	<i>1.4</i>	<i>0.59</i>	1	10
CP	6.2	7.6	8.2	6.7	8.1	8.2	7.2	9.2	8.8	1	10
$\rho = 0.3$											
MILP	<i>1.1</i>	<i>0.96</i>	<i>0.63</i>	<i>1.3</i>	<i>1.1</i>	<i>0.57</i>	<i>1.2</i>	<i>1.0</i>	<i>0.59</i>	1	10
CP	5.9	6.8	7.5	6.5	7.5	7.8	6.5	7.7	8.05	1	10

**Table 3** Simplex iterations performed by the MILP approach (MILP-SI) and nodes explored by the CP approach (CP-Nod). Italic figures in the table highlight the approach that produced the best run time

$\bar{L}$	1500			2500			3500			$T$	$N$
	$a$	500	750	1000	500	750	1000	500	750		
$\rho = 0.15$											
MILP-SI	<i>14122</i>	<i>13383</i>	<i>8071</i>	<i>15868</i>	<i>13483</i>	<i>6666</i>	<i>15350</i>	<i>13854</i>	<i>7083</i>	1	10
CP-Nod	83480	103442	108266	83480	103442	108266	83480	103372	108266	1	10
$\rho = 0.3$											
MILP-SI	<i>16084</i>	<i>13460</i>	<i>8381</i>	<i>16937</i>	<i>13649</i>	<i>6683</i>	<i>15357</i>	<i>13854</i>	<i>7083</i>	1	10
CP-Nod 1	83480	103442	108266	83480	103442	108266	83480	103442	108266	1	10

3900, that is three orders of magnitude. Although MILP performs better at proving optimality, its performances are still far from those achieved by the CP approach. In fact the discrepancy between the two approaches with respect to feasible problems reaches a factor of 88: almost two orders of magnitude.

### 5.2 Discussion of results

The results presented indicate that the CP approach is in general more tractable than the mathematical programming one for this class of scheduling problems. The average solution time over all the instances considered is 950 seconds for the MILP approach and 24 seconds for the CP approach. This shows that, on average, CP is about one order of magnitude faster than MILP for the test bed analyzed. A comparison of solution times for the test problems reveals that, as the value of  $T$  increases (Fig. 7), CP is orders of magnitude faster than MILP, irrespectively of  $N$ ,  $\bar{L}$  and  $\rho$ . It should be noted that CP, as a consequence of constraint propagation, is extremely good at proving infeasibility, while this is the class of problems for which MILP requires significant computational efforts. CP also shows a more stable behavior and scalable performances as  $T$ ,  $N$ ,  $\bar{L}$  and  $\rho$  increase. MILP performs poorly for the largest instances considered both in proving optimality and detecting infeasibility.

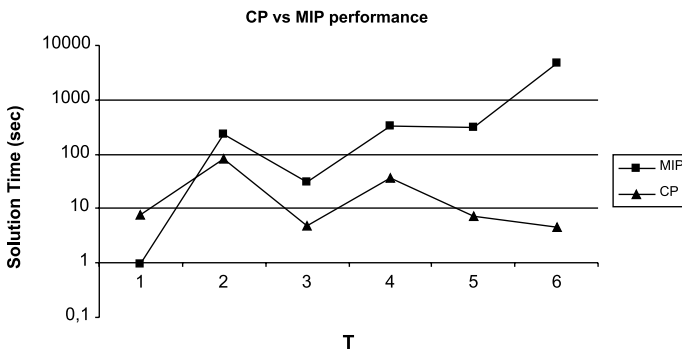
**Table 4** Computational times (in sec) for the MILP approach (MILP) and for the CP approach (CP). *Italic figures in the table highlight the approach that produced the best run time. Underlined figures are infeasible instances. +30000 means that the search has been stopped before infeasibility could be proved, after 30000 sec (8,3 hour)*

$\bar{L}$ <i>a</i>	1500			2500			3500			<i>T</i>	<i>N</i>
	500	750	1000	500	750	1000	500	750	1000		
$\rho = 0.15$											
MILP	150	120	98	240	240	200	310	420	310	2	20
CP	<i>9.4</i>	<i>9.6</i>	<i>9.6</i>	<i>92</i>	<i>110</i>	<i>130</i>	<i>110</i>	<i>120</i>	<i>170</i>	2	20
MILP	<u>22</u>	<u>17</u>	<u>41</u>	37	25	26	36	30	48	3	20
CP	<u>0.12</u>	<u>0.11</u>	<u>0.10</u>	7.6	7.3	7.4	6.4	6.8	7.6	3	20
MILP	<u>20</u>	<u>110</u>	<u>80</u>	500	320	400	610	540	610	4	30
CP	<u>0.07</u>	<u>0.07</u>	<u>0.07</u>	8.3	7.9	8.4	100	100	100	4	30
MILP	<u>0</u>	<u>0</u>	<u>0</u>	<u>700</u>	<u>1500</u>	<u>810</u>	250	140	420	5	30
CP	<u>0.06</u>	<u>0.07</u>	<u>0.06</u>	<u>1.0</u>	<u>1.0</u>	<u>1.0</u>	20	20	21	5	30
MILP	<u>0</u>	<u>0</u>	<u>0</u>	<u>2300</u>	<u>1500</u>	<u>1900</u>	660	1200	1500	6	40
CP	<u>0.11</u>	<u>0.11</u>	<u>0.10</u>	<u>0.45</u>	<u>0.39</u>	<u>0.41</u>	17	20	17	6	40
$\rho = 0.3$											
MILP	70	100	87	190	420	200	340	400	260	2	20
CP	<i>4.3</i>	<i>4.12</i>	<i>4.3</i>	<i>87</i>	<i>120</i>	<i>130</i>	<i>100</i>	<i>140</i>	<i>130</i>	2	20
MILP	<u>17</u>	<u>12</u>	<u>36</u>	22	31	26	36	34	33	3	20
CP	<u>0.08</u>	<u>0.08</u>	<u>0.08</u>	6.2	7.2	6.9	6.5	6.8	7.4	3	20
MILP	<u>39</u>	<u>20</u>	<u>23</u>	160	370	550	550	580	620	4	30
CP	<u>0.06</u>	<u>0.06</u>	<u>0.06</u>	5.0	4.9	5.2	110	110	120	4	30
MILP	<u>0</u>	<u>0</u>	<u>0</u>	<u>400</u>	<u>310</u>	<u>320</u>	300	130	440	5	30
CP	<u>0.07</u>	<u>0.07</u>	<u>0.07</u>	<u>0.77</u>	<u>0.77</u>	<u>0.77</u>	20	20	20	5	30
MILP	<u>0</u>	<u>0</u>	<u>0</u>	<u>1600</u>	<u>1500</u>	<u>700</u>	<u>20000</u>	<u>24000</u>	<u>+30000</u>	6	40
CP	<u>0.1</u>	<u>0.1</u>	<u>0.1</u>	<u>0.35</u>	<u>0.39</u>	<u>0.47</u>	<u>7.7</u>	<u>7.7</u>	<u>7.7</u>	6	40

The performance of CP-based and MILP-based approaches for solving a number of combinatorial optimization problems has been the scope of many recent studies (e.g., the modified generalized assignment problem, Darby-Dowman et al. 1997; the template design problem, Proll and Smith 1998; the progressive party problem, Smith et al. 1995). There has been effort to characterise the properties of different problems by their effect on the performance of CP and MILP approaches (Darby-Dowman et al. 1997; Jain and Grossmann 2001). The key result of that work is that MILP is very efficient when the relaxation is tight and the models have a structure that can be effectively exploited, while CP seems to work better for highly constrained discrete optimization problems in which the expressiveness of MILP is a major limitation. Our results confirm that the best model of choice depends on the characteristics of the instances rather than of the structure exposed at the problem level. Our experiments suggest that when the audit time is small the relaxation is tight, hence MILP performs well; when the audit time gets longer the problem becomes more constrained, and CP seems to scale up much better than MILP.

**Table 5** Simplex iterations performed by the MILP approach (MILP-SI) and Nodes explored by the CP approach (CP-Nod). *Italic figures in the table highlight the approach that produced the best run time. Underlined figures are infeasible instances. +59652116 means that the search has been stopped before infeasibility could be proved, after 30000 sec (8,3 hour)*

$\bar{L}$	1500			2500			3500			$T$	$N$
$a$	500	750	1000	500	750	1000	500	750	1000		
$\rho = 0.15$											
MILP-SI	737530	534238	444055	1115993	1102108	952999	1548908	2182999	1346656	2	20
CP-Nod	<i>70246</i>	<i>75317</i>	<i>80386</i>	<i>636450</i>	<i>764147</i>	<i>911016</i>	<i>636450</i>	<i>764147</i>	<i>911134</i>	2	20
MILP-SI	<u>115666</u>	<u>107912</u>	<u>227602</u>	191912	133002	137485	179505	150434	204205	3	20
CP-Nod	<u>511</u>	<u>511</u>	<u>511</u>	48025	51003	54077	48025	51003	54077	3	20
MILP-SI	<u>23915</u>	<u>176870</u>	<u>122716</u>	1745261	1056964	1285394	1842453	1580697	1565614	4	30
CP-Nod	<u>90</u>	<u>90</u>	<u>90</u>	56456	56526	56620	517734	538358	559576	4	30
MILP-SI	<u>p.</u>	<u>p.</u>	<u>p.</u>	<u>1889563</u>	<u>4362836</u>	<u>2365598</u>	889056	484259	1493608	5	30
CP-Nod	<u>26</u>	<u>26</u>	<u>26</u>	<u>6017</u>	<u>6017</u>	<u>6017</u>	104956	107308	109671	5	30
MILP-SI	<u>p.</u>	<u>p.</u>	<u>p.</u>	<u>3068784</u>	<u>2184342</u>	<u>2979721</u>	1440912	2280890	3245663	6	40
CP-Nod	<u>17</u>	<u>17</u>	<u>17</u>	<u>1358</u>	<u>1358</u>	<u>1358</u>	94461	94465	94467	6	40
$\rho = 0.3$											
MILP-SI	358921	455248	425701	902365	1946679	965218	1477861	2142193	1347372	2	20
CP-Nod	<i>38004</i>	<i>39394</i>	<i>40679</i>	<i>636450</i>	<i>764147</i>	<i>911134</i>	<i>636450</i>	<i>764147</i>	<i>911134</i>	2	20
MILP-SI	<u>105962</u>	<u>80243</u>	<u>210615</u>	117851	174027	129448	181685	165411	177510	3	20
CP-Nod	<u>511</u>	<u>511</u>	<u>511</u>	48025	51003	54077	48025	51003	54077	3	20
MILP-SI	<u>62566</u>	<u>29487</u>	<u>30184</u>	492571	1145508	1771707	1470525	1691075	1694070	4	30
CP-Nod	<u>90</u>	<u>90</u>	<u>90</u>	33660	33622	33622	511950	532192	553003	4	30
MILP-SI	<u>p.</u>	<u>p.</u>	<u>p.</u>	<u>1317658</u>	<u>951745</u>	<u>1052849</u>	1226532	444214	1482423	5	30
CP-Nod	<u>26</u>	<u>26</u>	<u>26</u>	<u>4885</u>	<u>4885</u>	<u>4885</u>	97399	99383	101332	5	30
MILP-SI	<u>p.</u>	<u>p.</u>	<u>p.</u>	<u>2267185</u>	<u>2616084</u>	<u>884031</u>	<u>34210413</u>	<u>46569011</u>	<u>+59652116</u>	6	40
CP-Nod	<u>17</u>	<u>17</u>	<u>17</u>	<u>1198</u>	<u>1198</u>	<u>1198</u>	<u>35354</u>	<u>35354</u>	<u>35354</u>	6	40



**Fig. 7** Comparison of the average solution time for the CP approach and for the MILP approach as a function of the audit time  $T$

## 6 Conclusions

This paper addresses the stochastic combinatorial optimization problem of scheduling internal audit activities. In Tarim et al. (2008) a related problem has been tackled by means of a similar MILP approach, but the authors assume that audit activities are instantaneous (conducting an audit does not take any time). Our work is more general and more realistic since we consider non-instantaneous audit activities, and we schedule the audit team in such a way as to prevent concurrent auditing.

We proposed a stochastic programming formulation and we developed two alternative certainty equivalent approaches to solve this model: an MILP model and a CP model. Our computational experience shows that MILP proved to be effective when the time required to perform an audit is short ( $T \leq 1$ ). In contrast, our CP approach proved to be very effective when the audit time  $T$  is greater than one period. The CP approach proved extremely effective both in proving optimality and detecting infeasibility for most of the instances considered. For instances where the audit time  $T$  is greater than one, the CP approach proves optimality or detect infeasibility in a time that is typically orders-of-magnitude less than the one required by the MILP approach. Nevertheless the performance of the CP approach when the audit time is short still remains acceptable and close to that achieved by the MILP approach.

Finally, we believe that introducing additional complexity in the model may constitute an interesting direction for future research. For instance, heterogeneous audit teams may be considered, which may take different times to perform audits; alternatively, random audit durations—rather than a fixed and deterministic duration  $T$ —may be incorporated in the stochastic programming model.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## Appendix

In this [Appendix](#) a complete list of the notation adopted in the paper is given.

- $N$ : (constant) number of time periods in the planning horizon
- $i, j, t$ : (index) a time period
- $M$ : (constant) number of audit units
- $m$ : (index) an audit unit
- $l_t^m$ : (random variable) a normally distributed random variable representing losses that accrue in audit unit  $m$  during period  $t$
- $g_t^m(l_t^m)$ : (function) probability density function of  $l_t^m$
- $E[.]$ : (function) expected value operator
- $\mu_t^m$ : (constant) expected value of  $l_t^m$ , sometimes expressed as  $E[l_t^m]$
- $\sigma_t^m$ : (constant) standard deviation of  $l_t^m$
- $\rho$ : (constant) coefficient of variation of  $l_t^m$ ,  $\rho = \sigma_t^m / \mu_t^m$

$T$ :	(constant) number of time periods required by the audit team to complete an audit
$a$ :	(constant) the fixed cost incurred each time an audit is conducted
$h$ :	(constant) the loss discount factor measuring the opportunity cost associated with a given loss level
$\bar{L}$ :	(constant) a threshold indicating the maximum allowed loss level in each period
$\alpha$ :	(constant) the probability of not exceeding the loss threshold $\bar{L}$
$L_t^m$ :	(decision variable) the loss level in audit unit $m$ at the beginning of period $t$
$E[TC]$ :	(objective function) the sum of expected audit costs and discounted period losses that are expected to accrue in the absence of auditing
$K_t^m$ :	(decision variable) a variable that takes the value of 1 if an internal audit (lasting $T$ periods) is started for audit unit $m$ in period $t$ , otherwise 0
$\mathfrak{M}$ :	(constant) some very large number
$A_r^m$ :	(index) time period in which the $r$ th audit is performed on unit $m$
$G_{l_t^m}(x)$ :	(function) $G_{l_t^m}(x) = \int_{-\infty}^x g_{l_t^m}(\tau) d\tau$ is the cumulative distribution function of $l_t^m$
$G_{l_t^m}^{-1}(\alpha)$ :	(function) the inverse cumulative distribution function (or $\alpha$ -quantile) of $l_t^m$
$G_N(\cdot)$ :	(function) the standard normal distribution function
$G_N^{-1}(\cdot)$ :	(function) the inverse of $G_N(\cdot)$
$P_{t,j}^m$ :	(decision variable) a binary variable that takes a value of 1 if the most recent audit prior to period $t$ was in period $j$ and zero elsewhere for a given audit unit $m$
$C$ :	(constant) maximum number of audit teams that the firm can assign to conduct audits in any given time period
$\Phi[m, t, j]$ :	(constant table) a 3-dimensional table whose elements are defined as $\Phi[m, t, j] = \bar{L} - G_{l_j^m + l_{j+1}^m + \dots + l_t^m}^{-1}(\alpha)$ , $t = 1, \dots, N$ , $j = 1, \dots, N$

**References**

Apt K (2003) Principles of constraint programming. Cambridge University Press, New York  
 Birge JR, Louveaux F (1997) Introduction to stochastic programming. Springer, New York  
 Bookbinder JH, Tan JY (1988) Strategies for the probabilistic lot-sizing problem with service-level constraints. *Manag Sci* 34:1096–1108  
 Boritz JE, Broca DS (1986) Scheduling internal audit activities. *Auditing A J Pract Theory* 6:1–19  
 Brailsford SC, Potts CN, Smith BM (1999) Constraint satisfaction problems: Algorithms and applications. *Eur J Oper Res* 119:557–581  
 Charnes A, Cooper WW (1959) Chance-constrained programming. *Manag Sci* 6(1):73–79  
 Chvtal V (1983) Linear programming. Freeman, New York  
 Colmerauer A (1985) Prolog in 10 figures. *Commun ACM* 28(12):1296–1310

- Dantzig GB (1951) Maximization of a linear function of variables subject to linear inequalities. In: Activity analysis of production and allocation. Wiley, New York (Chap XXI)
- Dantzig GB (1963) Linear programming and extensions. Princeton University Press, Princeton
- Darby-Dowman K, Little J, Mitra G, Zaffalon M (1997) Constraint logic programming and integer programming approaches and their collaboration in solving an assignment scheduling problem. *Constraints* 1(3):245–264
- Hanus M (1998) Programming with constraints: An introduction by K Marriott and PJ Stuckey. MIT Press, Cambridge. See also *J Funct Program* 11(2):253–262 (2001)
- Hughes J (1977) Optimal internal audit timing. *Account Rev* 52:56–68
- Ilog (2007a) CPLEX 9.0 Users Manual. ILOG Inc, Incline Village, NV
- Ilog (2007b) OPL Studio 3.7 Users Manual. ILOG Inc, Incline Village, NV
- Jain V, Grossmann IE (2001) Algorithms for hybrid milp/cp models for a class of optimization problems. *INFORMS J Comput* 13:258–276
- Jeffreys H (1961) Theory of probability. Clarendon, Oxford
- Knechel WR, Benson HP (1991) The optimization approach for scheduling internal audits of division. *Decis Sci* 22:391–405
- Lustig IJ, Puget JF (2001) Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces* 31:29–53
- Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley-Interscience, New York
- Proll L, Smith B (1998) Integer linear programming and constraint logic programming approaches to a template design problem. *INFORMS J Comput* 10:265–275
- Regin J-C (1994) A filtering algorithm for constraints of difference in CSPs. In: Proceedings of the national conference on artificial intelligence (AAAI-94), Seattle, WA, USA, pp 362–367
- Regin J-C (2003) Global constraints and filtering algorithms. In: Milano M (ed) Constraints and integer programming combined. Kluwer, Dordrecht
- Schrijver A (1986) Theory of linear and integer programming. Wiley, New York
- Smith B, Brailsford S, Hubbard P, Williams HP (1995) The progressive party problem: integer linear programming and constraint programming compared. In: CP95: Proceedings of the 1st international conference on principles and practice of constraint programming, Marseilles
- Tarim SA, Smith B (2008) Constraint programming for computing non-stationary ( $R, S$ ) inventory policies. *Eur J Oper Res* 189(3):1004–1021
- Tarim SA, Tayfur O, Karacaer S (2008) A mip model for scheduling multi-location internal audit activities. Technical report, Hacettepe University, Dept of Management
- Tsang EPK (1993) Foundations of constraint satisfaction. Academic Press, London/San Diego
- Van Hentenryck P, Carillon JP (1988) Generality vs. specificity: an experience with AI and or techniques. In: Proceedings of the national conference on artificial intelligence (AAAI-88)
- Van Hentenryck P (1989) Constraint satisfaction in logic programming. MIT Press, Cambridge
- Van Hentenryck P (1999) The OPL optimization programming language. MIT Press, Cambridge
- Ventsel ES (1979) Theory of probability. Nauka, Moscow (in Russian)
- Williams HP (1994) Model building in mathematical programming, 3rd edn. Wiley, New York
- Wilson D, Ranson R (1971) Internal audit scheduling—a mathematical model. *The Internal Auditor*, No. July–August, pp. 42–50