Algorithmic Operations Research Vol.4 (2009) 155–170

A Two-Period Portfolio Selection Model for Asset-backed Securitization

Renata Mansini

University of Brescia, Department of Electronics for Automation, via Branze 38, I-25123 Brescia, Italy

Ulrich Pferschy

University of Graz, Department of Statistics and Operations Research, Universitätsstr. 15, A-8010 Graz, Austria

Abstract

Asset-Backed Securitization (ABS) is a well-stated financial mechanism which allows an institution (either a commercial bank or a firm) to get funds through the conversion of assets into capital market products called notes or asset-backed securities. In this paper, we analyze the combinatorial problem faced by the financial institution which has to optimally select the set of assets to be converted into notes. We assume that assets follow an amortization rule characterized by constant periodic principal installments (Italian amortization). The particular shape of the assets outstanding principal is exploited both in the mathematical formulation of the problem and in its solution. In particular, we study a model formulation for the special case where assets selection occurs at two dates during the securitization process. We introduce two heuristic approaches based on Lagrangian relaxation and analyze their worst-case behavior compared to the optimal solution value. The performance of the algorithms is tested on a large set of problem instances generated according to two real-world scenarios provided by a leasing company. The proposed approximation algorithms turn out to yield solutions of high quality within very short computation time. The comparison to the solution approach applied by practitioners yields an average improvement of roughly 10% of the objective function value.

Key words: Asset-Backed Securitization, Italian amortization, Lagrangian heuristics, worst-case analysis

1. Introduction

Asset-Backed Securitization (ABS) is a device of structured financing where an institution, typically a commercial bank, pools assets with identifiable cash flows over time, transfers the same to investors usually with the support of further financial entities, and thereby achieves the purpose of financing. Specifically, the cash flows of the assets are identified, consolidated and separated from the originating institution, and then broken into marketable instruments of fixed denomination (*notes* or *asset-backed securities*) to be offered to investors. To ensure marketability, the instruments must have general acceptability, i.e. they are either rated by credit rating agencies or they are secured by charge over substantial assets. Further, to ensure liquidity, the instruments are generally made in homogenous lots.

An ABS process usually involves several financial institutions. The institution that securitizes its assets is called the *Originator*. Through an ABS process, the Originator isolates some of its assets (typically, lease or mortgage assets) from the rest of its business and it hives them off in exchange for a long term loan (main outstanding principal) to a financial institution called *Special Purpose Vehicle* (SPV). The SPV acts as a housing device, i.e. it houses the assets transferred by the Originator in a legal outfit and issues the notes related to them. This financial institution is created with the pure and special aim of issuing notes and its life is destined to end when the purpose is attained. Finally, the notes are purchased by institutional investors (usually commercial banks) which sell them to *final investors*. Issued notes yield an interest payable on periodic bases and are divided into tranches characterized by different maturity dates. The reimbursement to the holders of a tranche of notes corresponds to a reimbursement installment of the main outstanding principal. Hence, the outline of the outstanding principal of the loan has as many installments (steps) as the number of tranches of notes with different maturity issued on the market. Further, issued securities are related to the specific assets and to their cash flows: the final investors receive periodic in-flows (interests on their investments) which are

Email: Renata Mansini [rmansini@ing.unibs.it], Ulrich Pferschy [pferschy@uni-graz.at].

directly related to the periodic installments paid by the holders of the assets to the Originator.

The securitization technique arises in all those practical contexts where receivables form a large part of the total assets of an institution. Besides, to be packaged as a security, the ideal receivable is one which is repayable over or after a certain period of time, and there is contractual certainty as to its payment. Hence, its application has been principally directed towards housing/mortgage and leasing/hire-purchase companies (in the USA as well as in Europe the market for securities based on mortgage and leasing receivables forms a substantial part of total securitization markets), car rental companies, credit cards companies, etc. More recently, companies from fields such as electricity, telecommunication, aviation and insurance have joined as users of securitization. For a detailed description of the whole ABS process, the role of financial entities involved and for an analysis of the reasons which make it a profitable financial tool we refer to [7] and to the monographs cited therein.

In the past, ABS has been analyzed qualitatively from a financial perspective. Only in the last decade some papers dealing with ABS in terms of mathematical formulations have appeared. In [4] and [5] pre-payment models and a mean-absolute deviation portfolio optimization for mortgage-backed securities are presented. In [8] and then in [9] the Originator's problem of optimally selecting the assets to be converted into notes is studied for the first time. The problem was motivated by the real case of a leasing company where the outstanding principal of the assets follows French amortization rule, i.e. it is characterized by constant general (principal plus interests) installments. The authors make the assumption that assets can be selected at a unique date: the resulting problem is modelled as a multidimensional knapsack problem which is hardly tractable by exact algorithms but is typically solved by constructive heuristics or metaheuristics (see e.g. [2]). They also show that in the special case where all lease assets share the same financial characteristics (amortization rule, internal interest rate and term), all but one constraint turn out to be redundant and the model reduces to a classical, relatively easy to handle, 0-1 Knapsack Problem (see [6]). In these works the authors do not take into account the possibility of a different rule for the amortization of assets and do not analyze the case of selecting assets at different dates which frequently occurs in practice.

In the present paper we innovate with respect to the previous literature by using the Italian amortization as rule for assets and main outstanding principal amortization. We motivate our study with the practical need of finding alternative and possibly more effective formulations for the problem of selecting assets in an ABS process. Moreover, the particular shape of the outstanding principal based on constant principal installments (Italian amortization rule) allows to introduce a more realistic model where assets can be selected in more than one date during the ABS duration, provided that the whole set of assets is available at the initial date of the process. In particular, we further extend the analysis started in [7] by investigating the scenario in which exactly two points are available for selection of assets: the closing date and one more point in time. In [7] the authors analyze the problem of optimally selecting assets at multiple dates by comparing Italian with French amortization and propose four approximation algorithms based on LP-relaxation and on the implicit knapsack structure of the problem. Their extensive computational analysis has put in evidence that the selection of assets at the closing date of the ABS process is of critical importance while the successive dates provide a much lower contribution to the objective function. This has motivated the analysis of the special case studied in this paper. Our problem objective is, as in [9] and [7], to minimize at each time the difference between the sum of the outstanding principals of the selected assets and the outstanding principal of the main loan. Such a gap has a precise financial meaning as described in [7].

The contribution of this paper is twofold. From a combinatorial optimization point of view, we introduce a new model for the problem of optimally selecting assets to be converted into notes. The special structure of the proposed model is then exploited to set forth Lagrangian upper bounds and solution algorithms. Finally, the worst-case performance of the proposed procedures is analyzed and compared to their average performance on random instances based on real case scenarios. From a financial point of view, the paper provides useful insights for financial institutions involved with asset-backed securitization processes. Moreover, the proposed two-period model helps the Originator in better formalizing the problem and its objective function, whereas Lagrangian heuristics can be used as practical tools extremely competitive and easy to implement. The effectiveness of these algorithms is underlined by a direct comparison with the heuristic approach used by practitioners to solve ABS problems and the best heuristic algorithm proposed in [7]. In the former case, the results of this comparison are impressive: Indeed, our heuristics improve the objective function value, i.e. the sum of the outstanding principals of the selected assets, by roughly 10% in average and never less than by 5%, which may result in huge savings for the company involved in the ABS process.

The paper is organized as follows: Section 2. describes the combinatorial problem faced by an Originator when optimally selecting the portfolio of assets to be converted into notes and introduces the two-period model where assets can be selected at the closing date of the ABS process and at another successive specified date. In Section 3. Lagrangian relaxation is applied to the proposed model providing two upper bounds used as starting points to construct feasible solutions. Lagrangian heuristic procedures are described in Section 4. where worst-case performance ratios are also provided. Extensive computational results on test problems randomly generated according to two real-world scenarios where receivables are leasing assets are presented in Section 5.. In the same section, we compare the Lagrangian heuristics behavior with a greedy approach typically used in practice to solve the problem and with the best heuristic (i.e. ABS-Knap) proposed in [7]. Finally, concluding remarks and future developments are given in Section 6..

2. The two-period portfolio selection problem

The Originator's main problem is that to select a set of assets in such a way that the sum of their outstanding principals never exceeds, at any point in time, the outstanding principal of the loan received by the SPV. Actually, in order to maximize its gain the Originator has to select assets to minimize the gap between the main outstanding principal and the outstanding principal of the selected assets over all points in time. Figure 1 shows an example of main outstanding principal and its area partially covered by the sum of the outstanding principals of two assets handed-over at the closing date (time t_0). The gap between the two profiles measures a loss of profit due to missing more profitable investments with higher yields.

In the following we introduce a model formulation for the case in which only two dates for asset selection are considered. The first one is the closing date t_0 (the starting date of the main loan) while the second one is a successive date t before main loan deadline. Thus, any asset i may be selected either at the closing date (time t_0) or at a single specified later time t. We will refer to these two dates as *settlement dates* while the assets

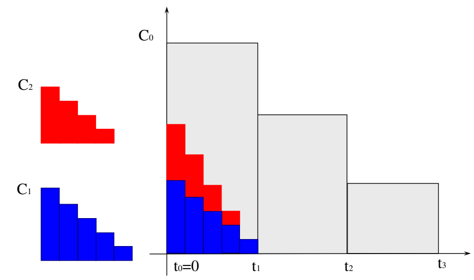

Fig. 1. The problem of asset selection in an ABS process.

given out by the Originator at such dates will be collectively referred to as *portfolio of assets*. We assume that all assets are made available at the initial date of the securitization and that, although such model can be used for any type of amortization rule chosen by the Originator, we analyze the case of Italian amortization exploiting the convenient features this amortization variant provides.

Let us consider a discretized time horizon $0, 1, 2, \ldots$ where the time unit is the periodicity of the reimbursement of the assets which is assumed to be the same for all the assets. The reimbursement dates for the main loan are assumed to happen at multiples r of this time unit. In practice time unit is equal to one month and r usually ranges between 18 and 24 months. Let $D := \{t_1, \ldots, t_T\}$ be the set of reimbursement dates of the main loan, where $t_1 > 0$. Let us define as n, $n = r \cdot T$, the main loan duration and as C its initial value. Since $t_0 = 0$, n corresponds to the date of the main loan last reimbursement t_T. We assume $t \in D$.

We indicate as M, with $|M| = m$, the set of assets available at the closing date. Note that M may include assets underwritten at the closing date as well as assets that have been underwritten before the closing date and the outstanding principal of which has been partially reimbursed. Each asset i, $i \in M$, is characterized by an initial value C_i of its outstanding principal which corresponds to the value of the asset outstanding principal at the closing date, i.e. $C_i = C_{i0}$, and by the number of its principal installments n_i which is equivalent to its duration expressed in time units and to its maturity date.

We assume that the assets are sorted in non-increasing order of their duration, i.e. $n_1 \geq n_2 \geq \ldots \geq n_m$ and that $\{1, \ldots, m_1\} = \{i \mid n_i > t\}$, with $m_1 \leq m$, i.e. the first m_1 assets expire after the selection date t. The outstanding principal of either assets or main loan reduces over time with constant principal installments according to the Italian amortization rule. To evaluate the advantages of using this amortization rule with respect to the French one we refer to [7].

Let us define as C_{i1} the outstanding principal of asset i at time t and as p_{i0} and p_{i1} the sum of outstanding principals of asset i, from time 0, resp. time t, to the asset maturity n_i:

$$C_{i1} = \frac{C_i}{n_i}(n_i - t),$$
$$p_{i0} = \frac{C_i}{2}(n_i + 1),$$
$$p_{i1} = \frac{C_i}{2n_i}(n_i - t)(n_i - t + 1).$$

If $i > m_1$, C_{i1} and p_{i1} are equal to zero. The value of the main loan outstanding principal at time t is given by $C - \frac{C}{n}t$.

We introduce two sets of binary variables indicating the selection of each asset i at the initial date t_0 and at time t, as follows:

$$x_{i0} = \begin{cases} 1 & \text{if asset } i \text{ is included in the portfolio at time } t_0, \\ 0 & \text{otherwise;} \end{cases}$$

$$x_{i1} = \begin{cases} 1 & \text{if asset } i \text{ is included in the portfolio at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $x_{i1} = 0$ for $i > m_1$.

The problem of selecting the assets at time t_0 and t in such a way that the sum of their outstanding principals never exceeds the outstanding principal of the main loan can be formulated as a linear integer program as follows:

$$\text{(ABS-2)} \quad v(t) := \max \sum_{i=1}^{m} p_{i0}x_{i0} + \sum_{i=1}^{m_1} p_{i1}x_{i1} \qquad (1)$$
$$x_{i0} + x_{i1} \le 1, \qquad i = 1, \dots, m, \qquad (2)$$
$$\sum_{i=1}^{m} C_i x_{i0} \le C, \qquad (3)$$
$$\sum_{i=1}^{m_1} \frac{C_i}{n_i}(n_i - t)(x_{i0} + x_{i1}) \le C - \frac{C}{n}\,t\,, \qquad (4)$$
$$x_{i0}, x_{i1} \in \{0, 1\}, \qquad i = 1, \dots, m\,. \qquad (5)$$

The objective function $v(t)$ establishes the maximization of the sum of the outstanding principals of the assets from the date of their selection to their maturity. This is equivalent to minimizing the gap between the main loan outstanding principal and the sum of the outstanding principals of the assets over time. Moreover, since both main loan and assets follow Italian amortization rule the sum of outstanding principals of the selected assets will never exceed the main outstanding principal provided that it does not exceed the loan at the dates of assets selection (see [7]). The set of constraints (2) implies that each asset can be selected at most once either at time 0 or at time t. Constraint (3) states that the sum of the outstanding principals of the assets selected at time t_0 must not exceed the outstanding principal of the main loan at the same date. Similarly, constraint (4) establishes that the sum of the outstanding principals of the assets selected before and at time t must not exceed the outstanding principal of the main loan at the same date.

Problem (ABS-2) is a generalization of a two-dimensional knapsack problem with additional XOR–conditions between pairs of variables and a special structure of the data. The standard two-dimensional knapsack problem was treated extensively in [10], where the authors also considered its Lagrangian relaxation in detail. If all assets expire before time t the problem reduces to a simple knapsack problem. Hence, $m_1 > 0$ will be assumed. The same problem using French instead of Italian amortization rule would have implied a multi-dimensional knapsack problem requiring an additional constraint for each reimbursement date of the main outstanding principal (see also [7]). The relatively simple structure of (ABS-2) with only two capacity constraints can be exploited by a Lagrangian relaxation approach to find upper bounds on the optimal solution value and to create feasible suboptimal portfolios. This will be the objective of the next section.

3. Lagrangian Relaxation

Instead of directly exploiting the implicit knapsack structure of the problem we propose a theoretically more advanced approach based on the application of Lagrangian relaxation to Problem (ABS-2). The computation of upper bounds based on Lagrangian relaxation is aimed at the extraction of feasible solutions, thus it is not promising to relax both constraints at the same time removing most of the structure of the problem. Instead, we will pursue the two possibilities of relaxing either constraint (3) or constraint (4) yielding the two parametric problems $P_1(\lambda)$ and $P_2(\lambda)$. For both cases will we consider the Lagrangian dual problems DP_1

and DP_2. Each of the resulting solution values gives an upper bound on $v(t)$ and the corresponding solution frequently provides a reasonable selection of assets. Thereby, one can hope for a better solution at the cost of a more complex computation. In general, this solution may be not feasible for the original problem and can only be transformed heuristically into a feasible solution. In our approach this will fortunately not be necessary. Theory on Lagrangian relaxation and its application to combinatorial optimization problems can be found in the seminal paper [3] or in the textbook [11].

Let us start with relaxing constraint (3).

$$DP_1 \qquad \min_{\lambda \geq 0} \quad f_1(\lambda),$$

where $f_1(\lambda)$ is defined as the solution value of the following problem:

$$P_1(\lambda) \qquad f_1(\lambda) := \max \sum_{i=1}^{m} p_{i0} x_{i0} + \sum_{i=1}^{m_1} p_{i1} x_{i1} + \lambda \left(C - \sum_{i=1}^{m} C_i x_{i0} \right) \quad (6)$$

$$x_{i0} + x_{i1} \leq 1, i = 1, \ldots, m, \quad (7)$$

$$\sum_{i=1}^{m_1} \frac{C_i}{n_i}(n_i - t)(x_{i0} + x_{i1}) \leq C - \frac{C}{n}\, t\,, \quad (8)$$

$$x_{i0}, x_{i1} \in \{0, 1\}, \qquad i = 1, \ldots, m\,. \quad (9)$$

It is well known that $f_1(\lambda)$ is an upper bound for $v(t)$ for every λ since the optimal solution of (ABS-2) is trivially feasible for $P_1(\lambda)$. The objective function $f_1(\lambda)$ can be rewritten as

$$\max \sum_{i=1}^{m} r_i(\lambda) x_{i0} + \sum_{i=1}^{m_1} p_{i1} x_{i1} + \lambda C \quad (10)$$

with

$$r_i(\lambda) := p_{i0} - \lambda C_i\,.$$

$P_1(\lambda)$ can be modelled by a standard knapsack problem KP where an "item" i corresponds to the selection of a pair x_{i0}, x_{i1} represented by the better alternative. Weights and profits are given as $w_i := \frac{C_i}{n_i}(n_i - t)$ and $p_i(\lambda) := \max\{p_{i0} - \lambda C_i, p_{i1}\}$, $i = 1, ..., m_1$, whereas the knapsack capacity is trivially $C - \frac{C}{n}\, t$. Naturally, assets with duration shorter than t are not included in the knapsack instance but selected if and only if $p_{i0} - \lambda C_i > 0$:

$$P_1'(\lambda) \qquad \max \quad \sum_{i=1}^{m_1} p_i(\lambda) z_i$$

$$\sum_{i=1}^{m} \frac{C_i}{n_i}(n_i - t) z_i \leq C - \frac{C}{n} t\,,$$

$$z_i \in \{0, 1\}, i = 1, \ldots, m_1\,.$$

For a given value λ_0, denoting the optimal solution of $P_1'(\lambda_0)$ by z_i^* we can compute the value of $f_i(\lambda_0)$ as follows: if $z_i^* = 1$ we set $x_{i0}(\lambda_0) := 1$ if $r_i(\lambda_0) \geq p_{i1}$ and $x_{i1}(\lambda_0) := 1$, otherwise. If $z_i^* = 0$ both variables are set to zero.

Notice that in the knapsack problem $P_1'(\lambda)$, every profit $p_i(\lambda)$ is a convex function consisting of a linear function starting with $p_i(0) = p_{i0}$ and decreasing with slope $-C_i$ until $\lambda = (p_{i0} - p_{i1})/C_i$, from where the profit continues as a constant function $p_i(\lambda) = p_{i1}$ as depicted in Figure 2 (left side).

The second possibility for a Lagrangian relaxation of (ABS-2) is to relax constraint (4) yielding:

$$DP_2 \qquad \min_{\lambda \geq 0} f_2(\lambda),$$

where $f_2(\lambda)$ is defined as the solution value of the following problem:

$$P_2(\lambda) \qquad f_2(\lambda) := \max \sum_{i=1}^{m} p_{i0} x_{i0} + \sum_{i=1}^{m_1} p_{i1} x_{i1} + \lambda \left(C - \frac{C}{n} t - \sum_{i=1}^{m_1} \frac{C_i}{n_i}(n_i - t)(x_{i0} + x_{i1}) \right)$$

$$x_{i0} + x_{i1} \leq 1, i = 1, \ldots, m, \quad (11)$$

$$\sum_{i=1}^{m} C_i x_{i0} \leq C, \quad (12)$$

$$x_{i0}, x_{i1} \in \{0, 1\}, i = 1, \ldots, m\,. \quad (13)$$

As before, $f_2(\lambda)$ is a trivial upper bound for $v(t)$ for every λ. Constraint (3) should be more significant for the solution value (cf. remarks at the end of the paper [7]). Hence, one can hope that DP_2, which guarantees that (3) is fulfilled, should generate solutions only moderately violating (4).

In the case of DP_2, the objective function yielding $f_2(\lambda)$ can be equivalently written as:

$$\sum_{i=1}^{m} a_i(\lambda)\; x_{i0} + \sum_{i=1}^{m_1} b_i(\lambda)\; x_{i1} + \lambda \left(C - \frac{C}{n}\, t \right) \quad (14)$$

with

$$a_i(\lambda) := \max\left\{0, p_{i0} - \lambda\,\frac{C_i}{n_i}(n_i - t)\right\},$$
$$b_i(\lambda) := \max\left\{0, p_{i1} - \lambda\,\frac{C_i}{n_i}(n_i - t)\right\},$$

for $n_i > t$, and

$$a_i(\lambda) := p_{i0}, \quad b_i(\lambda) := 0,$$

otherwise.

For a given argument λ_0 the computation of $f_2(\lambda_0)$ works as follows. Note that the selection of x_{i1} does not cause any effect in constraint (12). Hence, according to (14), an optimal solution always exists where both variables x_{i0}, x_{i1} will never be set equal to 0 but exactly one of them will be set to 1 as along as $a_i(\lambda) > 0$. Since $a_i(\lambda) \geq b_i(\lambda)$, the corresponding decisions can be modeled by the following standard knapsack problem, where $y_i = 1$ indicates that $x_{i0} = 1$ and $x_{i1} = 0$.

$$P_2'(\lambda) \qquad \max \sum_{i=1}^{m} (a_i(\lambda) - b_i(\lambda))y_i$$
$$\sum_{i=1}^{m} C_i\, y_i \leq C\,,$$
$$y_i \in \{0,1\} \qquad i = 1, \ldots, m\,.$$

Denoting the optimal solution of $P_2'(\lambda_0)$ by y_i^* we can immediately deduct the value of $f_2(\lambda_0)$ in (14) by setting $x_{i0}(\lambda_0) := y_i^*$ and $x_{i1}(\lambda_0) := 1 - y_i^*$.

Taking a closer look at the coefficients $a_i(\lambda)$ and $b_i(\lambda)$ we can characterize the profit values $p_i(\lambda) := a_i(\lambda) - b_i(\lambda)$ in the above knapsack problem $P_2'(\lambda)$ as follows. For small λ, both $a_i(\lambda)$ and $b_i(\lambda)$ are strictly positive and their difference $p_{i0} - p_{i1}$ is constant for every i. Hence, the solution of a single instance of KP is an optimal solution for $P_2'(\lambda)$ as long as $b_i(\lambda) > 0$ holds for all i. For the case where λ yields $a_i(\lambda) > 0$ but $b_i(\lambda) = 0$, the KP instance has profits linearly decreasing in λ. For larger λ, there is also $a_i(\lambda) = 0$ and item i can be discarded from consideration. The situation is illustrated in Figure 2 (right side). Of course, the breakpoints of λ separating these three cases are different for every i. In the special case $n_i \leq t$, there is $p_i(\lambda) = p_{i0}$.

4. Lagrangian Heuristics

It is well known and easy to see that $f_i(\lambda)$, $i = 1, 2$, is a piecewise linear convex function over $\mathbf{R}^+$. In

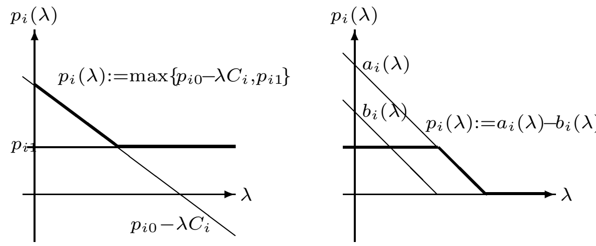

Fig. 2. The profits of the knapsack problems associated with DP_1 (left figure) and DP_2 (right figure) as functions of λ.

fact, in both cases DP_1 and DP_2 the solution value of every feasible solution is a piecewise linear convex function in λ, and hence the property follows for $f_i(\lambda)$. However, it should be noted that the solution function of $P_2'(\lambda)$, which is used for computing values of $f_2(\lambda)$, is piecewise linear and decreasing but not at all convex.

It was pointed out before that in general a Lagrangian relaxation approach will not necessarily generate a feasible solution for an integer programming problem. However, in the case of DP_1 and DP_2 the objective functions $f_i(\lambda)$ have a special structure. In particular, the convex functions $f_i(\lambda)$ consist of a decreasing part with $\lambda < \lambda_{\min}$ for which the corresponding solutions are infeasible for the original problem (ABS-2) since the slope is given by the slack of the relaxed constraint. For $\lambda > \lambda_{\min}$ the slopes of the linear pieces are positive and correspond to feasible solutions with a positive slack in the constraint. The position $\lambda_{\min}$ (excluding degenerate cases where a linear piece has slope 0) is a breakpoint of $f_i(\lambda)$. It is defined by the intersection of the linear piece with maximal negative slope and the piece with minimal positive slope. Taking the feasible solution corresponding to the latter of these pieces provides the feasible solution of our heuristic. We will denote the resulting heuristics as HL_1 and HL_2.

4.1. *Searching for the Lagrangian Dual*

Computing the minimum of $f_i(\lambda)$ can be done by a variety of general search techniques. In all cases, we have at hand during the execution of such an algorithm a search interval $[\lambda^\ell, \lambda^u]$ containing the optimal multiplier $\lambda_{\min}$. By means of the specific search procedure a new multiplier $\lambda_{new} \in (\lambda^\ell, \lambda^u)$ is computed. After evaluating $f_i(\lambda_{new})$ we set λ_{new} as a new lower or upper interval bound depending on the sign of the slope of $f_i(\lambda)$ at λ_{new}.

We will apply three different approaches to determine a new multiplier:

(i) *Bisection:* This trivial procedure performing binary search on the λ-axis simply chooses $\lambda_{bis} := (\lambda^\ell + \lambda^u)/2$ as depicted in Figure 3. It does not take into account any available information on $f_i(\lambda)$.

(ii) *Outer Approximation:* Considering the slopes of $f_i(\lambda)$ in the two endpoints of the search interval we can intersect the corresponding tangents and thus find a new search value λ_{oa} as the λ-coordinate of the intersection point (see Figure 3).

(iii) *Outer Approximation with Angle Bisection:* If $f_i(\lambda)$ is fairly symmetric w.r.t. λ_{min}, Outer Approximation should perform very well. For highly asymmetric shapes of $f_i(\lambda)$ a modification of this method immediately comes to mind. Instead of the intersection point of the two tangents we construct the bisecting line of their angle. It is not unlikely that this line points roughly towards the minimum of $f_i(\lambda)$. To get a new multiplier, we intersect this bisecting line with a horizontal line given by any lower bound on $f_i(\lambda)$. Obviously, every feasible solution for (ABS-2) (derived by a heuristic) gives such a lower bound.

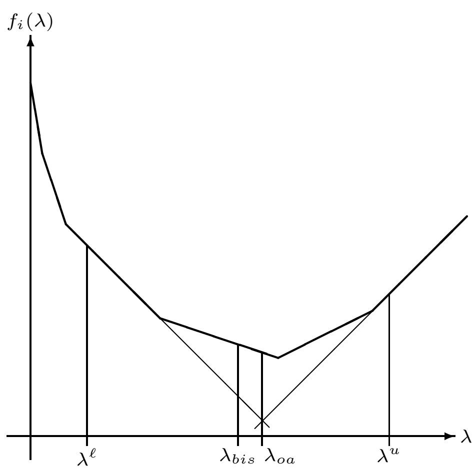

Fig. 3. Search step for a given interval $[\lambda^\ell, \lambda^u]$: Bisection yields $\lambda_{bis} = (\lambda^\ell + \lambda^u)/2$, Outer Approximation intersects the two tangents of $f_i(\lambda)$ at λ^ℓ and λ^u producing λ_{oa}.

4.2. *Initialization and Stopping Criteria*

Important issues for the performance of the Lagrangian heuristics are the choice of the initial search interval and the applied stopping criterion. Both of these features were shown to be crucial elements in practical experiments with Lagrange-type heuristics.

The standard choice of starting the search for λ_{min} with $\lambda^\ell := 0$ can not be easily improved. Simple initial upper bounds on λ_{min} can be found by taking a closer look at the specific objective functions.

For $f_1(\lambda)$ we can easily compute a multiplier λ^u as

$$\lambda^u := \max_{i=1,\ldots,m} \frac{p_{i0} - p_{i1}}{C_i}. \qquad (15)$$

For this choice of λ^u there is $r_i(\lambda) \le p_{i1}$ for all i in (10). Hence, our function $f_1(\lambda)$ has a fixed positive slope of C at λ^u. Performing elementary calculations for (15) it follows immediately that the maximum is attained for the largest n_i, i.e. for $i = 1$, which results in

$$\lambda^u = \frac{p_{10} - p_{11}}{C_1} = t\left(1 - \frac{t-1}{2n_1}\right).$$

Note that $\lambda^u \in (t/2\,, t)$. Elaborating further on this approach it can be seen that the slope of $f_1(\lambda)$ will stay positive as long as $C > \sum_{i=1}^m C_i x_{i0}$. This can be guaranteed also for a smaller multiplier, namely if λ is chosen such that the sum of C_i of those assets i where $r_i(\lambda) > p_{i1}$ is less than C. More formally, determine

$$i' := \arg \max_{i=1,\ldots,m} \sum_{\ell=1}^{i} C_\ell < C,$$

and compute as above an upper bound

$$\lambda^u = \frac{p_{i'0} - p_{i'1}}{C_{i'}} = t\left(1 - \frac{t-1}{2n_{i'}}\right). \qquad (16)$$

Recall that the assets are assumed to be sorted in decreasing order of duration.

In the case of $f_2(\lambda)$ one can follow similar considerations and compute a multiplier such that all $a_i(\lambda)$ are 0. This yields

$$\lambda^u := \max_{i=1,\ldots,m} \frac{p_{i0}\, n_i}{C_i(n_i - t)} = \max_{i=1,\ldots,m} \frac{n_i(n_i+1)}{2(n_i - t)}. \qquad (17)$$

Contrary to the previous case, the expression to be maximized (if treated as a continuous instead of pointwise function) is not monotonous in n_i but attains a local minimum for $n_i = t + \sqrt{t^2 + t}$. Hence, the above maximum may be attained either for the asset with longest or with shortest duration, which results in

$$\lambda^u = \max\left\{\frac{n_1(n_1+1)}{2(n_1 - t)}\,,\ \frac{n_m(n_m+1)}{2(n_m - t)}\right\}.$$

The same arguments applied above to derive (16) can be also used to improve the value of λ^u for $f_2(\lambda)$. However, we have to take care of the discussed behaviour of the expression maximized in (17). This means that we gradually decrease λ^u by collecting a set of assets S in the following way: Take the asset which generates the maximum in (17), put it into S and temporarily remove it from consideration (only during the computation of the initial upper bound on λ_{min}). Recompute λ^u as defined in (17) with the reduced set of assets and iterate this procedure as long as

$$C - \frac{C}{n}t > \sum_{i \in S} \frac{C_i}{n_i}(n_i - t).$$

Upon termination this process will have implicitly computed two indices i' and i'' such that $S = \{1, \ldots, i', i'', \ldots, m\}$ (one of the two intervals may be empty). Taking λ^u as the maximum over S instead of $\{1, \ldots, m\}$ will yield $a_i(\lambda) > 0$ only for assets $i \in S$. By the above stopping criterion even selecting all these assets will produce a positive slope of $f_2(\lambda)$ at λ^u.

Standard stopping criteria are bounds on the width of the remaining search interval such as $\lambda^u - \lambda^\ell < \delta_1$ or on the positive value of the slope associated with the solution at λ^u, which is equivalent to an upper bound on the slack in the relaxed constraint of (ABS-2). The only way to determine $\lambda_{\min}$ with certainty is based on the Outer Approximation described above. Indeed, the intersection point of the two linear pieces of the function $f_i(\lambda)$ always lies below the actual function $f_i(\lambda)$. As soon as the evaluation of $f_i(\lambda)$ at this point yields the same solution set, i.e. the same linear piece, as in the previous iteration, we know that $\lambda_{\min}$ is equal to the λ–coordinate of the intersection point. Of course, this property also holds for Outer Approximation with Angle Bisection.

A related setup in the area of knapsack problems involving a search procedure on a piecewise linear, convex function, where every search value corresponds to the solution of a standard knapsack problem, was treated in [1] for the different context of an inverse-parametric knapsack problem. Some of the computational features reported in their work can also be used for the present problems.

4.3. *Worst-Case Analysis*

Concerning the performance of the two Lagrange-based heuristics we can give theoretical results showing that both heuristics based on $P_1(\lambda)$ and $P_2(\lambda)$ can behave as bad as possible.

Example 1 *Consider a long term loan with normalized volume $C = 1$, a given parameter ℓ and the second selection point $t = \ell - 1$. Let three assets be given with the following data:*

i	1	2, 3
C_i	1	$\frac{1}{2} + \varepsilon$
n_i	ℓ	ℓ
p_{i0}	$\frac{1}{2}(\ell + 1)$	$(\frac{1}{4} + \frac{\varepsilon}{2})(\ell + 1)$
p_{i1}	$\frac{1}{\ell}$	$(\frac{1}{2} + \varepsilon)\frac{1}{\ell}$
C_{i1}	$\frac{1}{\ell}$	$(\frac{1}{2} + \varepsilon)\frac{1}{\ell}$

The duration of the main loan is set to $n = (\ell - 1)/(1-(1+2\varepsilon)1/\ell)$ thus guaranteeing that asset 2 and 3 can be selected together at time t since $C_{01} = (1+2\varepsilon)\frac{1}{\ell}$.

There are six feasible solutions (excluding symmetry) to this example: Selecting the single asset 1 or 2 at time 0 or time t, selecting asset 2 at time 0 and asset 3 at t and finally selecting both assets 2 and 3 at time t. In the Lagrangian relaxation $P_1(\lambda)$ we get as additional feasible solution the selection of both assets 2 and 3 at time 0 and the empty set.

Constructing the Lagrangian solution functions for each of these eight cases yields the following scenario represented by the corresponding non-zero variables:

function	*selection*	*profit*	λ *– slope*	*dominated by*
$g_1(\lambda)$	x_{10}	p_{10}	0	
$g_2(\lambda)$	x_{11}	p_{11}	1	$g_6(\lambda)$
$g_3(\lambda)$	x_{20}	p_{20}	$\frac{1}{2} - \varepsilon$	$g_5(\lambda)$
$g_4(\lambda)$	x_{21}	p_{21}	1	$g_6(\lambda)$
$g_5(\lambda)$	x_{20}, x_{31}	$p_{20} + p_{31}$	$\frac{1}{2} - \varepsilon$	
$g_6(\lambda)$	x_{21}, x_{31}	$p_{21} + p_{31}$	1	
$g_7(\lambda)$	x_{20}, x_{30}	$p_{20} + p_{30}$	-2ε	
$g_8(\lambda)$	$\emptyset$	0	1	$g_6(\lambda)$

Figure 4 illustrates these eight solutions as linear functions in λ. Those functions which almost coincide are represented only by a single line. The upper envelope $f_1(\lambda)$, i.e. the maximum over all functions, is given in bold.

Observing the above definitions it can be easily seen that $g_6(\lambda)$ dominates $g_2(\lambda)$, $g_4(\lambda)$ and $g_8(\lambda)$ which can thus be eliminated from further consideration. Moreover, $g_5(\lambda)$ dominates $g_3(\lambda)$. Among the remaining four functions we can calculate that $g_5(\lambda)$, $g_6(\lambda)$ and $g_7(\lambda)$ share a single intersection point at $\bar{\lambda} = \frac{p_{20} - p_{21}}{C_2} = \frac{1}{2}(\ell + 1) - \frac{1}{\ell}$. From the slope of the functions it can be seen that $g_7(\lambda) > g_5(\lambda)$ for $\lambda < \bar{\lambda}$ and $g_6(\lambda) > g_5(\lambda)$ for $\lambda > \bar{\lambda}$. Thus, $g_5(\lambda)$ cannot be part of $f_1(\lambda)$. Finally,

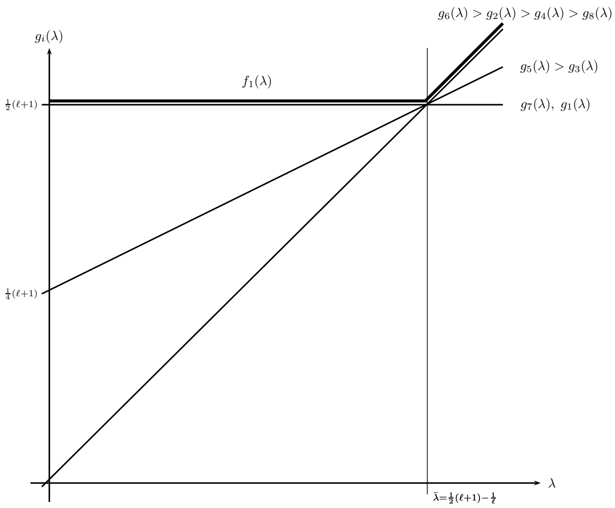

Fig. 4. Lagrange functions of the eight feasible solutions of Example 1.

an elementary calculation shows that $g_1(0) < g_7(0)$ *and* $g_1(\bar{\lambda}) < g_7(\bar{\lambda}) = g_6(\bar{\lambda})$. *Considering again the slopes of the three functions we have established:*

$$f_1(\lambda) = \begin{cases} g_7(\lambda) & \text{for } \lambda \leq \bar{\lambda}, \\ g_6(\lambda) & \text{for } \lambda > \bar{\lambda}. \end{cases}$$

Hence, HL_1 *outputs as a solution the selection of both assets* 2 *and* 3 *but only at time* t, *whereas the optimal solution chooses only asset 1 at time 0. Therefore, the performance ratio of* HL_1 *can be bounded from above by*

$$\lim_{\ell\to\infty} \frac{HL_1}{Opt} = \lim_{\ell\to\infty} \frac{p_{21}+p_{31}}{p_{10}} = \lim_{\ell\to\infty} \frac{(1+2\varepsilon)\frac{1}{\ell}}{\frac{1}{2}(\ell+1)} = 0.$$

Theorem 1 *The performance ratio of heuristic* HL_1 *can be arbitrarily bad.* □

Example 2 *Consider again a long term loan with normalized volume* $C = 1$ *and duration equal to* n. *The second selection point is set to* $t = 1$. *Let three assets be given with the following data (*$\varepsilon < 1$*).*

i	*1*	2	3
C_i	1	1	ε
n_i	1	$n-1$	n
p_{i0}	1	$\frac{n}{2}$	$\varepsilon \cdot \frac{n+1}{2}$
p_{i1}	0	$\frac{n-2}{2}$	$\varepsilon \cdot \frac{n-1}{2}$
C_{i1}	0	$1-\frac{1}{n-1}$	$\varepsilon \cdot \frac{n-1}{n}$

Excluding the trivially dominated solutions consisting of only asset 2 or 3 selected at time $t = 1$, *there are six feasible solutions: Selecting any single asset at time 0 and nothing else, combining asset 1 selected at time 0 with one of the other assets selected at time 1 and the empty set. In the Lagrangian relaxation* $P_2(\lambda)$ *we get three additional feasible solutions, namely the selection of assets 2 and 3 one at time 0 the other at time 1 (which clearly dominates selecting both of them only at time 1) and selecting asset 1 at time 0 and both assets 2 and 3 at time 1. The resulting Lagrangian solution functions for all nine relevant cases are given in table 1.*

For all three infeasible solutions the slope of the Lagrangian function is given by $h(\varepsilon) := \frac{1}{n(n-1)} - \varepsilon \cdot \frac{n-1}{n}$ *which is negative for* $\varepsilon > \frac{1}{(n-1)^2}$. *Note that some of the dominations are not strict but they can be obtained by appropriate tie-breaking rules. The nine solutions are illustrated as linear functions in* λ *in Figure 5. As before functions with tiny deviation are represented only by a single line and the maximum over all functions is given in bold.*

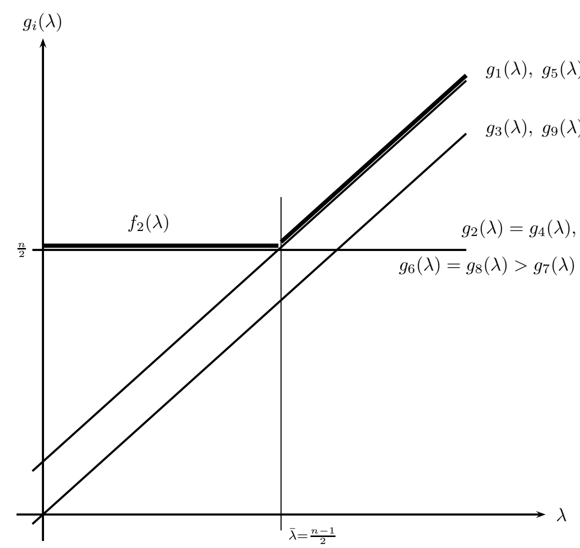

Fig. 5. Lagrange functions of the eight feasible solutions of Example 2.

To determine the upper envelope $f_2(\lambda)$ *we start by sorting the four remaining Lagrangian solution functions for* $\lambda = 0$ *which yields* $g_6(0) > g_2(0) > g_5(0) > g_1(0)$. *Furthermore, for* λ *tending to infinity there is* $f_2(\lambda) = g_1(\lambda)$. *Now we compute the intersection point of* $g_1(\lambda)$ *and* $g_2(\lambda)$ *denoted by* $\bar{\lambda}$. *An elementary calculation yields* $\bar{\lambda} = \frac{n-1}{2}$.

Further calculations show that $g_5(\bar{\lambda}) < g_1(\bar{\lambda})$ *(for*

Table 1

function	selection	profit	λ-slope	dominated by
$g_1(\lambda)$	x_{10}	p_{10}	$1-\frac{1}{n}$	
$g_2(\lambda)$	x_{20}	p_{20}	$\frac{1}{n(n-1)}$	
$g_3(\lambda)$	x_{30}	p_{30}	$(n-1)(\frac{1}{n}-\frac{\varepsilon}{n})$	$g_5(\lambda)$
$g_4(\lambda)$	x_{10}, x_{21}	$p_{10}+p_{21}$	$\frac{1}{n(n-1)}$	$g_2(\lambda)$
$g_5(\lambda)$	x_{10}, x_{31}	$p_{10}+p_{31}$	$(n-1)(\frac{1}{n}-\frac{\varepsilon}{n})$	
$g_6(\lambda)$	x_{20}, x_{31}	$p_{20}+p_{31}$	$h(\varepsilon)$	
$g_7(\lambda)$	x_{30}, x_{21}	$p_{30}+p_{21}$	$h(\varepsilon)$	$g_6(\lambda)$
$g_8(\lambda)$	x_{10}, x_{21}, x_{31}	$p_{10}+p_{21}+p_{31}$	$h(\varepsilon)$	$g_6(\lambda)$
$g_9(\lambda)$	$\emptyset$	0	$1-\frac{1}{n}$	$g_1(\lambda)$

$n > 3$) which implies that the intersection point of $g_1(\lambda)$ and $g_5(\lambda)$ is smaller than $\bar{\lambda}$. Hence, $g_5(\lambda)$ is dominated by $g_2(\lambda)$ for $\lambda \le \bar{\lambda}$ and by $g_1(\lambda)$ for $\lambda > \bar{\lambda}$ and will never be a part of $f_2(\lambda)$.

On the other hand it can be shown by a straightforward calculation that $g_6(\bar{\lambda}) > g_1(\bar{\lambda})$. Hence, the intersection point of $g_1(\lambda)$ and $g_6(\lambda)$ is larger than $\bar{\lambda}$. Therefore, we can conclude that $g_6(\lambda)$ dominates $g_2(\lambda)$ for $\lambda \le \bar{\lambda}$ and $g_1(\lambda)$ trivially dominates $g_2(\lambda)$ for $\lambda > \bar{\lambda}$. Hence, $g_2(\lambda)$ will never be a part of $f_2(\lambda)$ which can now be described as $\max\{g_6(\lambda), g_1(\lambda)\}$.

The Lagrangian heuristic HL_2 will thus output the feasible solution corresponding to $g_1(\lambda)$, i.e. selecting only asset 1 at time 0 which yields a profit of 1 whereas the optimal solution would select only asset 2 at time 0 reaching a profit of $\frac{n}{2}$:

$$\lim_{n\to\infty} \frac{HL_2}{Opt} = \lim_{n\to\infty} \frac{p_{10}}{p_{20}} = \lim_{n\to\infty} \frac{1}{\frac{n}{2}} = 0.$$

Thus we have shown the following theorem:

Theorem 2 *The performance ratio of heuristic HL_2 can be arbitrarily bad.* □

The two worst-case examples also show that there is no dominance between the two Lagrangian heuristics. It can be checked that heuristic HL_2 finds the optimal solution for Example 1 and heuristic HL_1 reports the optimal solution for Example 2. Without going into the details of the computation, this is due to the fact that in each example one of the constraints is only slightly violated by the solution given by the heuristic it is constructed to fail for. Applying the other heuristic, the required feasibility of that constraint immediately leads to the optimal solution.

Corollary 3 *There is no dominance between HL_1 and HL_2.* □

5. Computational Analysis

5.1. *Testing environment*

This section is devoted to a comprehensive analysis and comparison of the proposed Lagrangian heuristics in a real-case decision environment provided by a Leasing Bank in Milan (i.e. we assume that assets are leasing contracts and the Originator is the bank itself).

Data sets are generated according to the parameter ranges shown in Table 3 where the leasing contracts are classified, according to the type of the underlying asset, into 5 different classes. More precisely, each class is characterized by the type of assets involved, their economic value (initial value of the underlying asset) and by their average expiring date. An asset i belonging to the class A_s , $s = 1, ..., 5$, has the outstanding principal at the closing date C_i^s and the duration (i.e. number of installments) n_i^s uniformly generated in the ranges $[C^s_{min}, C^s_{max}]$ resp. $[n^s_{min}, n^s_{max}]$. For instance, class A_1 (cf. first line of Table 3) consists of lease contracts with value (initial outstanding principal) ranging between 5000 and 75,000 Euros whose underlying assets are small vans and motor vehicles. The asset duration (the term) for this class has an average length equal to 35 months and can range between 25 and 45 months. Notice that assets with larger initial values have longer life (longer average term).

The Leasing Bank provides us with two different realistic scenarios for generating data.

(1) Scenario S_1: $h_s := 1/5, \forall s.$
(2) Scenario S_2:
  h_3 uniformly distributed in $[0, 0.3]$
  h_4 uniformly distributed in $[0, 0.2]$
  h_5 uniformly distributed in $[0, 0.15]$
  $h_1 = h_2 := \frac{1}{2}(1 - h_3 - h_4 - h_5)$.

Scenario S_1 assumes that each class has the same number of assets, whereas scenario S_2 imposes an upper bound on the percentage h_s of the total number of

Table 2

Class	Underlying Asset	$[C^s_{min}, C^s_{max}]$	average term (in months)	$[n^s_{min}, n^s_{max}]$
A_1	vehicles & vans	$[5, 75]$	35	$[25, 45]$
A_2	plant & machinery	$[5, 75]$	48	$[38, 58]$
A_3	vehicles & trucks	$[75, 1500]$	47	$[37, 57]$
A_4	plant & machinery	$[75, 1500]$	54	$[44, 64]$
A_5	real estate	$[75, 2000]$	95	$[85, 105]$

Generation of assets for the classes A_1 to A_5.

assets which belong to each class s. For instance, the cardinality of class A_3 has to be lower than or equal to 30% of the total number of generated assets.

In real world problems the total number of assets available at the closing date may vary widely since the bank can carry out different securitizations. On average, each ABS operation involves thousands of assets. We have investigated realistic problems with four different values of m equal to 500, 1000, 5000 and 10000.

The time unit for asset reimbursement is equal to one month, while the reimbursement dates of the main loan are equally distributed over the ABS duration with interval length between two dates equal to 18 or 24 months and a number of reimbursement dates ranging between 4 and 6. More precisely, the main loan is assumed to have a duration n equal to 96 and 120 months (resp. 8 and 10 years) if the reimbursement dates occur every 24 months, while it lasts 90 and 108 months (resp. 7.5 and 9 years) when reimbursement dates have periodicity equal to 18 months. In all cases the main loan has a term longer than the longest available asset. We have set the initial value C of the main outstanding principal equal to $\frac{1}{2}\sum_{i=1}^{m} C_i$. Given n, the settlement date t is set equal to the first and the third reimbursement date of the main loan, respectively. For instance, if the term n of the main loan is equal to 96 months we have generate instances with t equal to 24 and 72 months, respectively. In particular, we have analyzed the following eight pairs of values (n,t): (96,24), (96,72), (120,24), (120,72), (90,18), (90,54), (108,18) (108,54). For each m and each pair (n, t) we have generated 5 instances with the same initial value of the main loan outstanding principal for scenario S_1 and 5 for scenario S_2. This means 80 instances for each m and 320 instances altogether.

Test problems have been run on a PC Intel(R) Pentium IV Windows 2000 with 1500 Mhz. The code has been written in Microsoft© Visual C++. Optimal solutions of (ABS-2) have been computed with CPLEX© Version 8.1. In particular, we have set a computation time limit of 2 hours for CPLEX branch and cut routine. This means that if no optimal solution has been found within this time limit, the routine is stopped and the best integer feasible solution found so far is provided.

5.2. *Computational Results*

In the following we provide the computational analysis and the comparison of the performances of the two proposed heuristics. To generate the multiplier λ we have implemented all the three methods described in Section 4.1.. Nevertheless, we have decided to report only the results for the Bisection (indicated as Bis) and the Outer Approximation (indicated as OutApp) methods. The performances of the Outer Approximation with Angle Bisection approach strongly depends on the solution value of the heuristic used as lower bound. We have used two different values for this bound. The first one is based on the solution of an instance of KP which results from setting $x_{i1} = 0$ for all i in (ABS-2), i.e. solving the problem with assets selected only at t_0. If the difference between the average duration of the assets and t is fairly large one may expect to find quite a good lower bound. Unfortunately this was not always the case in our test problems. We then decided to use as lower bound the final value provided by HL_1 resp. HL_2 when using the Bisection method. In all cases, the obtained computational performances were never better than those found with Outer Approximation, thus we decided to omit this approach from further consideration.

We start the discussion of the computational results by analyzing the efficiency and effectiveness of CPLEX in solving the 320 instances to optimality within 2 hours of computation time. Since solution values are of order larger than 10^9 Euro when $m = 500$, larger than 10^{10} when $m = 1000$ and 5000 and larger than 10^{11} when m =10000, we have changed the mixed integer optimality gap tolerance of CPLEX from its default value (equal to 10^{-4}) to 10^{-8}. In the case of instances with

$m = 10000$ CPLEX may terminate with a solution which differs from the optimal one by several thousand Euros. Obviously the time required for the proof of optimality increases when the gap tolerance is reduced.

Tables 4 and 5 provide the number of instances solved to optimality within the time threshold of two hours for Scenario S_1 and S_2, respectively. Each entry of the two tables shows the number of generated instances (in brackets) and the number of instances solved to optimality. About 80% and 75% of all instances of Scenario S_1 resp. S_2 have been solved to optimality within 2 hours of computation time. For the unsolved instances, we have tried to remove the threshold on the computation time finding out that, for almost all of them, CPLEX was unable to find the optimal solution even after 10 hours of computation time. By comparing the two tables its evident that on average the number of solved instances is quite similar in the two scenarios. In particular, in both cases, the most difficult problems are those selecting assets at the first reimbursement date of the main loan (see columns (96,24) (120,24), (90,18) and (108,18)) rather than selecting assets at the third one. The difference in terms of the computation time required is impressive: none of the problems with t equal to the third reimbursement date of the main loan has required CPLEX more than 15 seconds to find the optimal solution with an average time equal to 1.39 seconds. On the contrary, if we consider the problems solved to optimality with t equal to the first reimbursement date of the main loan (i.e. instances with (n, t) values equal to (96,24) (120,24), (90,18) and (108,18)) the average time required by CPLEX increases to 236.1 seconds with a maximum value equal to 2923 seconds. The result suggests that initial dates are most critical for asset selection.

For m = 500 all but a few instances could be solved to optimality. On the contrary, for some values of n and t (cf. (90,18) and (108,18) for Scenario S_2) no optimal solution was found after 2 hours for instances with 10000 assets.

Tables 6 and 7 show the average and the maximum computation time required by CPLEX and by the two heuristics to solve problems with the same number of assets. For CPLEX we provide either the average time out of the instances solved to optimality within 2 hours or that taking into account also those instances stopped after 2 hours (values into brackets). The running time of CPLEX to compute the optimal solution of (ABS-2) is on average quite high even for relatively small problems. The deviation between different problems of the same size is extremely large. Some instances could be solved in less than one second whereas others require hours.

For heuristics HL_1 and HL_2 most of the running time is consumed by the solution of knapsack problems for each value of λ. We have used CPLEX to solve $P_1'(\lambda)$ and $P_2'(\lambda)$ knapsack problems by setting the same tolerance gap used for optimal solutions. Of course a specialized KP-code might be plugged in to further improve the performance. On average, the computation time increases with the number of assets and both heuristics are more efficient when using Bisection with respect to Outer Approximation method. This is especially true for heuristic HL_2 where the time required to solve the KP problems is, on average, higher with respect to HL_1 and the difference between Outer Approximation and Bisection is justified by the larger number of problems solved when using the first method with respect to the second one. As can be seen by comparing the two tables, no relevant differences can be noted between the two scenarios. Finally, if we take a closer look to the results it can be noticed that for heuristic HL_1, those instances with t set equal to the first reimbursement date of the main loan require, on average, higher computation times with quite large deviations among different instances with the same m.

To measure the quality of the approximate solutions we have computed their relative deviation from the optimal solution value, if available, or from the best integer solution value found by CPLEX after 2 hours. Table 8 shows the average and the maximum percentage errors found by heuristic HL_1 for each m under Scenario S_1 and S_2. The performances are shown for the cases with Bisection and with Outer Approximation. Table 9 provides similar results for heuristic HL_2.

The average error found by heuristic HL_1 changes only slightly with the number of assets, while its value is, on average, larger for instances belonging to Scenario S_2 with respect to those of scenario S_1. If we analyze the average performance of heuristic HL_1 with Outer Approximation (Bisection) on those instances with t equal to 18 and 24 (i.e. t equal to the first reimbursement date of the main loan) we find an average percentage gap from the optimal solution value equal to 0.021% (0.022%) for the instances in Scenario S_1 and to 0.11% (0.23%) for those in Scenario S_2. The corresponding values for instances with t equal to 54 and 72 are equal to 0.75% (0.85%) and 1.27% (1.43%) for Scenario S_1 and S_2, respectively. This can be justified by the fact that the constraint at time t is more relevant when t is closer to the closing date and that heuristic HL_1 obtained by

Table 3

	(96,24)	(96,72)	(120,24)	(120,72)	(90,18)	(90,54)	(108,18)	(108,54)	total
500	5 (5)	5 (5)	5 (5)	5 (5)	**4** (5)	5 (5)	**4** (5)	5 (5)	38 (40)
1000	5 (5)	5 (5)	**4** (5)	5 (5)	**4** (5)	5 (5)	**1** (5)	5 (5)	34 (40)
5000	**2** (5)	5 (5)	**2** (5)	5 (5)	**0** (5)	5 (5)	**2** (5)	5 (5)	26 (40)
10000	**1** (5)	5 (5)	5 (5)	5 (5)	**1** (5)	5 (5)	**2** (5)	5 (5)	29 (40)
total	13 (20)	20 (20)	16 (20)	20 (20)	9 (20)	20 (20)	9 (20)	20 (20)	127 (160)

Number of instances solved by CPLEX to optimality within 2 hours – Scenario S_1.

Table 4

	(96,24)	(96,72)	(120,24)	(120,72)	(90,18)	(90,54)	(108,18)	(108,54)	total
500	5 (5)	5 (5)	5 (5)	5 (5)	**4** (5)	5 (5)	**3** (5)	5 (5)	37 (40)
1000	**4** (5)	5 (5)	**3** (5)	5 (5)	**1** (5)	5 (5)	**1** (5)	5 (5)	29 (40)
5000	**4** (5)	5 (5)	**3** (5)	5 (5)	**2** (5)	5 (5)	**2** (5)	5 (5)	31 (40)
10000	**2** (5)	5 (5)	**2** (5)	5 (5)	**0** (5)	5 (5)	**0** (5)	5 (5)	24 (40)
total	15 (20)	20 (20)	13 (20)	20 (20)	7 (20)	20 (20)	6 (20)	20 (20)	121 (160)

Number of instances solved by CPLEX to optimality within 2 hours – Scenario S_2.

Table 5

m	CPLEX		HL_1				HL_2			
			Bis		OutApp		Bis		OutApp	
	av.	max.	av.	max.	av.	max.	av.	max.	av.	max.
500	97 (452)	2186	1.70	8	1.80	8	4.33	8	5.93	16
1000	162 (1218)	2923	2.75	15	2.95	14	8.82	21	13.98	43
5000	48 (2551)	1041	23.45	369	30.22	428	35.22	77	44.55	115
10000	72 (2032)	567	31.57	101	42.77	120	85.07	153	106.27	177

Average and maximum running time out of 40 instances – Scenario S_1.

Table 6

m	CPLEX		HL_1				HL_2			
			Bis		OutApp		Bis		OutApp	
	av.	max.	av.	max.	av.	max.	av.	max.	av.	max.
500	80 (614)	1128	1.38	4	1.23	5	5.45	25	8.83	22
1000	147 (2086)	3265	3.80	16	4.07	18	9.35	18	18.42	56
5000	30 (1643)	296	17.42	223	23.12	339	37.07	66	58.70	107
10000	78 (2927)	1707	32.85	120	41.12	152	88.15	166	114.55	171

Average and maximum running time out of 40 instances – Scenario S_2.

relaxing the constraint at the closing date is especially effective when the constraint at time t provides a larger contribution to the solution value.

Heuristic HL_2 shows a different behaviour since it is derived from Lagrangian relaxing the constraint at time t. Nevertheless, also this heuristic has a performance strongly dependent on the value of t, which tends to improve when t increases, i.e. when the relevance of the corresponding constraint decreases. Heuristic HL_2 has an average behaviour impressively better than that of heuristic HL_1. This is especially true in Scenario S_1. The result was somehow expected since HL_2 is based on the relaxation of the weaker of the two constraints of (ABS-2). The heuristic is able to find the optimal solution and even to improve the best integer solution provided by CPLEX in almost all the instances with t =72 and t =54. The result can be explained by the fact that these instances are characterized by a date t quite far from the closing date so that there is a limited number of assets with duration longer than t and the corresponding constraint provides a limited contribution to the solution value. However, also in those instances where t is set equal to 24 and 18 and where the corresponding constraint may provide a more relevant contribution to the solution value, heuristic HL_2 has an average performance for each m definitely better than heuristic HL_1. Finally, the negative percentage errors (see Table 9, Scenario S_1) mean that in many of those instances where the proof of optimality was not found after 2 hours of computation time, heuristics HL_2 has determined better integer solutions than those provided by CPLEX.

In conclusion, we can say that HL_2 is a highly successful algorithm which always yields the best solution quality. This result, supported by Table 9, is further strengthened by Table 10 which gives the number of instances where each algorithm has found the optimal solution value resp. the best known solution value. In addition, the numbers reported in brackets indicate how many times each heuristic has found a solution value

Table 7

	Scenario S_1				**Scenario** S_2			
	Bis		OutApp		Bis		OutApp	
m	av.	max.	av.	max.	av.	max.	av.	max.
500	0.483%	3.515%	0.547%	3.890%	1.389%	11.994%	0.848%	6.086%
1000	0.400%	5.336%	0.153%	1.673%	0.702%	5.714%	0.702%	5.714%
5000	0.285%	2.956%	0.275%	2.903%	0.575%	2.734%	0.575%	2.734%
10000	0.439%	3.637%	0.433%	3.576%	0.588%	1.916%	0.590%	1.916%

Heuristic HL_1. Average and maximum percentage errors – Scenario S_1 and S_2.

Table 8

	Scenario S_1				**Scenario** S_2			
	Bis		OutApp		Bis		OutApp	
m	av.	max.	av.	max.	av.	max.	av.	max.
500	0.047%	0.216%	-0.044%	0.216%	0.560%	11.297%	0.052%	0.447%
1000	0.021%	0.125%	-0.064%	0.125%	0.127%	1.963%	0.127%	1.963%
5000	0.011%	0.172%	-0.060%	0.039%	0.059%	1.466%	0.059%	1.466%
10000	-0.191%	0.017%	-0.191%	0.017%	0.003%	0.121%	0.003%	0.121%

Heuristic HL_2. Average and maximum percentage errors – Scenario S_1 and S_2.

strictly better than that provided by CPLEX after two hours.

In order to decide which method should be implemented for multiplier determination we can say that Outer Approximation method provides, on average, better results with respect to Bisection at the cost of higher computation times. In particular, the difference in computation time is relatively small for heuristic HL_1 so that we can recommend to use the Outer Approximation method with this algorithm, while for heuristic HL_2 the trade-off between computation time and quality of the solution suggests Bisection as the method of choice.

Finally, in [7] we investigate different heuristic algorithms to solve the multiple dates selection problem. Those heuristics can be easily modified and applied to the present problem with only two selection dates. In [7] the best performing heuristic was ABS-Knap. Applied to the present problem ABS-Knap requires the sequential optimal solution of a 0-1 Knapsack problem at each of the two selection dates. We tested this heuristic on all 320 instances of our data set. The heuristic is quite efficient never requiring more than a few minutes of running time. It is also quite effective reaching in the majority of instances the same solution as found by CPLEX in two hours, but never improving it. However, for a small number of instances the performance of ABS-Knap was quite bad with a deviation from the CPLEX solution of more than 10%.

In Table 11 we directly compare the performance of ABS-Knap with respect to the best Lagrangian method (i.e. HL_2(Out App)). Specifically, we report the maximum and average percentage gap of ABS-Knap with respect to procedure HL_2. A positive value means that ABS-Knap has a worse performance with respect to HL_2, whereas a negative one means the reverse is true. On average, ABS-Knap works quite well. Nevertheless, the Lagrangian approach clearly dominates for scenario S_1 with an improvement of up to 5%. For scenario S_2 the picture is less clear and no definite winner can be determined. However, the Lagrangian approach is to be preferred also for S_2 since it is the more stable method. Indeed, the maximum gap of the ABS-Knap solution can be more than 11% (cf. max gap values in Table 11), while for the instances where ABS-Knap yields better solution than the Lagrangian approach the gaps are fairly small, with a maximum deviation lower than 2%.

Table 10

	Scenario S_1		**Scenario** S_2	
m	av.	max.	av.	max.
500	0.041%	2.917%	0.462%	11.297%
1000	0.063%	1.171%	-0.129%	0.097%
5000	0.051%	1.729%	-0.052%	0.018%
10000	0.184%	4.351%	-0.004%	0.034%

HL_2 versus ABS-Knap. Average and maximum percentage errors – Scenario S_1 and S_2.

5.3. *What practitioners do*

The procedure currently applied in practice to select assets was communicated by managers from financial institutions dealing with the design of Asset-Backed Securities. This is a surprisingly simple greedy-type procedure: The assets are sorted in decreasing order of the value of their initial outstanding principal C_{i0}. Going through the list in this ordering an asset is added to the

Table 9

m	HL_1				HL_2			
	Bis		OutApp		Bis		OutApp	
	S_1	S_2	S_1	S_2	S_1	S_2	S_1	S_2
500	0	1 (1)	0	3	20 (2)	19 (3)	23 (4)	20 (1)
1000	1	1 (1)	3	1	20 (2)	21 (7)	25 (6)	21 (7)
5000	2	1	2	1	21	23 (1)	25 (3)	23 (1)
10000	3 (2)	1 (1)	3 (2)	1	24 (3)	24 (3)	24 (3)	24 (3)

Number of instances out of 40 solved to optimality or as good as CPLEX after 2 hours, number of instances with strictly better values in brackets.

Table 11

	Scenario S_1			**Scenario** S_2		
m	min.	av.	max.	min.	av.	max.
500	6.615%	10.067%	15.209%	5.140%	9.518%	21.060%
1000	7.075%	9.949%	15.150%	5.852%	9.399%	14.734%
5000	7.230%	9.494%	14.423%	6.148%	8.797%	12.968%
10000	4.839%	9.836%	13.964%	5.921%	9.347%	13.283%

Greedy-type heuristic applied in practice. Minimum, average and maximum percentage errors – Scenario S_1 and S_2.

portfolio if its initial outstanding principal together with the assets already selected does not violated the upper bound given by the outstanding principal of the main loan, otherwise it is discarded. After going through all assets and fixing the assets selected at t_0 an analogous process is performed for the selection at time t after re-ordering the items by their outstanding principal C_{i1}. This procedure corresponds in some sense to the classical greedy algorithm for the standard knapsack problem but does not take the asset duration into account.

Our computational results show that this real world selection process incurs a huge deviation from the solutions found by our heuristics HL_1 resp. HL_2 and by CPLEX after 2 hours. As can be seen from Table 12 this deviation is roughly 10% on average and still around 5% even in the best case! This comes as a surprise to some extent since greedy solutions perform bad (as can be seen by the usual worst-case analyses) if ”large” items are given filling a significant fraction of the available capacity. However, this is not the case in our test instances where all assets have only tiny principals compared to the main loan. We assume that the main reason for this bad performance is the applied sorting criterion which might be improved by using a product of the initial outstanding principal with the duration of the asset.

6. Conclusions and Future Developments

How should a financial manager face the challenging problem of optimally selecting the assets to be converted into notes? Do there exist effective formulations and solution methods for this problem? Practice has shown that a wrong selection of assets may result in a considerable loss of profit possibly making ineffective the whole financial operation. In this paper we provide useful insights for financial institutions involved with ABS processes. The proposed two-period model (ABS-2) helps decision makers to better formalize this complex problem and its objective function, whereas the analyzed Lagrangian heuristics can be used as practical methods extremely effective and easy to implement. Extensive computational results on a large set of problems generated accordingly to real world scenarios provided by a leasing bank show how the proposed approximation algorithms yield high quality solutions within very short computation time. Moreover, we notice that the solution of (ABS-2) depends heavily on the value of t. This moment in time will be in practice a point of negotiation between the seller and the purchaser and in many business scenarios it will be fixed only after the selection of contracts at time 0. Hence, the question arises for the seller (the bank) which selection date $t \in T := \{r, 2r, \ldots, \lfloor \frac{n}{r} \rfloor r\}$ would be most profitable w.r.t. $v(t)$. The resulting bi-level problem can be derived from (ABS-2) as

$$\max_{t \in T} \; v(t) \tag{18}$$

Clearly, this problem is even more difficult than (ABS-2) which we tried to tackle by heuristics. It is our intention to further analyze this problem from a theoretical point of view and provide the Originator with effective heuristics for solving it.

Acknowledgments: The authors wish to thank an anonymous reviewer for the useful comments which helped to improve the presentation of the work.

References

[1] Burkard, R.E., U. Pferschy. 1995. The inverse–parametric knapsack problem. *European Journal of Operational Research* **83** 376–393.

[2] Chu, P.C., J.E., Beasley. 1998. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics* **4** 63-86.

[3] Fisher, M.L. 1981. The Lagrangian relaxation method for solving integer programming problems. *Management Science* **27** 1–18.

[4] Kang, P., S.A. Zenios. 1992. Complete prepayment models for mortgage-backed securities. *Management Science* **38** 1665–1685.

[5] Kang, P., S.A. Zenios. 1993. Mean-absolute deviation portfolio optimization for mortgage-backed securities. *Annals of Operations Research* **45** 433–450.

[6] Kellerer, H., U. Pferschy, D. Pisinger. 2004. *Knapsack Problems*. Springer.

[7] Mansini, R., U. Pferschy. 2004. Securitization of Financial Assets: Approximation in Theory and Practice. *Computational Optimization and Applications* **29** 147–171.

[8] Mansini, R., M.G. Speranza. 2000. Selection of lease contracts in an asset-backed securitization: a real case analysis. *Control and Cybernetics* **28** 739–754.

[9] Mansini, R., M.G. Speranza. 2002. A multidimensional knapsack model for the Asset-Backed Securitization. *Journal of the Operational Research Society* **53** 822–832.

[10] Martello, S., P. Toth. 2003. An exact algorithm for the two-constraint 0–1 knapsack problem.*Operations Research* **51** 826–835.

[11] Wolsey, L.A. 1998. *Integer Programming*. Wiley.

Received 08-09-2008; revised 03-6-2009; accepted 01-09-2009