Proceedings of EQUADIFF 2017
pp. 79–88

THE TREE-GRID METHOD WITH CONTROL-INDEPENDENT STENCIL

IGOR KOSSACZKÝ , MATTHIAS EHRHARDT , AND MICHAEL GÜNTHER *

Abstract. The Tree-Grid method is a novel explicit convergent scheme for solving stochastic control problems or Hamilton-Jacobi-Bellman equations with one space dimension. One of the characteristics of the scheme is that the stencil size is dependent on space, control and possibly also on time. Because of the dependence on the control variable, it is not trivial to solve the optimization problem inside the method. Recently, this optimization part was solved by brute-force testing of all permitted controls. In this paper, we present a simple modification of the Tree-Grid scheme leading to a control-independent stencil. Under such modification an optimal control can be found analytically or with the Fibonacci search algorithm.

Key words. Tree-Grid Method, Hamilton-Jacobi-Bellman equation, Stochastic control problem, Fibonacci algorithm

AMS subject classifications. 65M75, 65C40

1. Introduction. Stochastic control problems (SCP) arise in many fields where some stochastic process is controlled in order to maximize (or minimize) an expected value of an uncertain outcome. An effective approach to solve such problems presents the Hamilton-Jacobi-Bellman (HJB) equation. As the analytical solutions are in most cases not feasible, the development of numerical methods dealing either with HJB equation or directly with the SCP is essential. A large class of methods is based on approximating the stochastic process by a Markov chain [5]. Another way presented e.g. in [2] is to solve the HJB equation with an implicit finite-difference method (FDM). A method based on Ricatti transformation of the HJB equation was proposed in [3].

Recently a new method having similarities with Markov chain approximations as well as with the explicit FDMs was presented in [4]. The advantage of this method is its independence on the space-stepping of the grid, as well as its unconditional convergence. However, as well as in FDMs and Markov chain methods, an optimization problem needs to be solved in each step.

In this paper, we want to present a modification of the Tree-Grid method, that will allow us to solve the optimization problem more effectively.

2. Problem formulation. The Tree-Grid method is a numerical scheme for searching the value function $V(s,t)$ of the following *general stochastic control problem*:

*University of Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany
igor.vyr@gmail.com, ehrhardt@math.uni-wuppertal.de, guenther@math.uni-wuppertal.de

$$
\begin{aligned}
V(s,t) = \max_{\theta(s,t)\in\bar{\Theta}} \mathbb{E}\Bigg(&\int_t^T \exp\left(\int_t^k r(S_l,l,\theta(S_l,l))dl\right) f(S_k,k,\theta(S_k,k))dk \\
&+ \exp\left(\int_t^T r(S_k,k,\theta(S_k,k))dk\right) V_T(S_T)\Big| S_t = s\Bigg), \qquad (2.1)
\end{aligned}
$$

$$
\begin{aligned}
dS_t =& \mu(S_t,t,\theta(S_t,t))dt + \sigma(S_t,t,\theta(S_t,t))dW_t, \qquad (2.2)\\
0 <& t < T, \quad s \in \mathbb{R},
\end{aligned}
$$

where s is the state variable and t denotes time. Here, $\bar{\Theta}$ is the space of all *suitable* control functions from $\mathbb{R} \times [0,T]$ to a set Θ. In the original Tree-Grid method [4], Θ is supposed to be discrete. If this is not the case, the set Θ should be discretized. Another option arising from this paper would be to search for an optimum analytically, that will be discussed later. Now following Bellman's principle, the following *dynamic programming equation* holds:

$$
\begin{aligned}
V(s,t_j) = \max_{\theta(s,t)\in\bar{\Theta}_{t_j}} \mathbb{E}\Bigg(&\int_{t_j}^{t_{j+1}} \exp\left(\int_{t_j}^k r(S_l,l,\theta(S_l,l))dl\right) f(S_k,k,\theta(S_k,k))dk \\
&+ \exp\left(\int_{t_j}^{t_{j+1}} r(S_k,k,\theta(S_k,k))dk\right) V(S_{t_{j+1}},t_{j+1})\Big| S_{t_j} = s\Bigg), \qquad (2.3)
\end{aligned}
$$

where $0 \le t_j < t_{j+1} \le T$ are some time-points and $\bar{\Theta}_{t_j}$ is a set of control functions from $\bar{\Theta}$ restricted to the $\mathbb{R} \times [t_j, t_{j+1})$ domain. Using this equation (2.3), it can be shown [7], that solving the SCP (2.1),(2.2) is equivalent to solving the so-called Hamilton-Jacobi-Bellman (HJB) equation:

$$
\begin{aligned}
&\frac{\partial V}{\partial t} + \max_{\theta\in\Theta}\left(\frac{\sigma(\cdot)^2}{2}\frac{\partial^2 V}{\partial s^2} + \mu(\cdot)\frac{\partial V}{\partial s} + r(\cdot)V + f(\cdot)\right) = 0, \qquad (2.4)\\
&V(s,T) = V_T(s), \qquad (2.5)\\
&0 < t < T, \quad s \in \mathbb{R},
\end{aligned}
$$

where $\sigma(\cdot)$, $\mu(\cdot)$, $r(\cdot)$, $f(\cdot)$ are functions of s, t, θ. This HJB formulation was used to prove the convergence of the scheme [4].

We should note that the maximum operator in (2.1) and (2.4) can be replaced by a minimum, (supremum, infimum) operator and the whole following analysis will hold analogously.

3. The Tree-Grid Method. The main idea of the Tree-Grid method is approximating the continuous stochastic process (2.2) with a discrete one, attaining only values from the grid inside the computational domain, or values outside the computational domain, that are assumed to be predefined. Then, a discretized version of (2.3) is used to compute the approximation of the value function in each node of the grid. The underlying discretized stochastic process can be easily represented by a scenario tree. However, such a tree is "growing" from every time-space node of an (arbitrarily chosen) grid, what explains the name of the method. We illustrate this structure in Figure 3.1. Alternatively, the method can be also interpreted in terms of finite differences which is discussed concisely in [4]. We will use this alternative representation also in the Sections 5, 6.

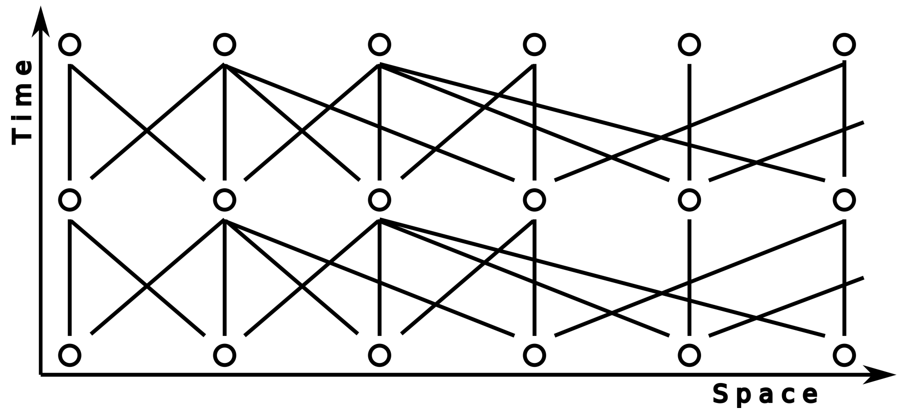

FIG. 3.1. *Illustration of the Tree-Grid structure. From each grid node in current time layer three branches are growing (bottom-to-top), determining which values from grid in later time layer influence the value in the current node.*

Now we will quickly recapitulate the Tree-Grid method algorithm. We compute the approximation of the solution on a rectangular domain $[s_L, s_R] \times [0, T]$ with some grid as in usual finite difference schemes for PDEs. The grid-points are denoted as $[s_i, t_j]$, $i \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, M\}$, $k < l \Rightarrow s_k < s_l, t_k < t_l$, $t_1 = 0, t_M = T$, $s_1 = s_L, s_N = s_R$. The grid is possibly non-equidistant in space with space-steps $\Delta_i s = s_{i+1} - s_i$ and $\Delta s = \max_i \Delta_i s$. We will use an equidistant discretization in time with a time-step Δt. A generalization to non-equidistant time-stepping is straightforward, however the implementation is less effective in means of computational time in that case. The numerical approximation of $V(s_i, t_j)$ will be denoted by v_i^j.

The whole scheme is then defined by the discrete approximation of the dynamic programming equation (2.3)

$$v_i^j = \max_{\theta \in \Theta} \Big(f_i^j(\theta)\Delta t + (1 + r_i^j(\theta)\Delta t) \cdot \Big(p_{(i-,\theta)} v_{(i-,\theta)}^{j+1} + p_{(i,\theta)} v_i^{j+1} + p_{(i+,\theta)} v_{(i+,\theta)}^{j+1} \Big)\Big). \tag{3.1}$$

for $i = 2, 3, \ldots, N-1$ and

$$v_1^j = BC_L(s_1, t_j), \quad v_N^j = BC_R(s_N, t_j). \tag{3.2}$$

Here, $f_i^j(\theta) = f(s_i, t_j, \theta)$, $r_i^j(\theta) = r(s_i, t_j, \theta)$ and

$$v_{(i*,\theta)}^{j+1} = \begin{cases} v_k^{j+1} \quad \text{so that } s_k = s_{(i*,\theta)} & \text{if } s_{(i*,\theta)} \in \{s_1, s_2, \ldots, s_N\} \\ BC_L(s_{(i*,\theta)}, t_{j+1}) & \text{if } s_{(i*,\theta)} < s_1 \\ BC_R(s_{(i*,\theta)}, t_{j+1}) & \text{if } s_{(i*,\theta)} > s_N \end{cases}$$

for the $* \in \{-, +\}$. Here $BC_L(s,t)$ and $BC_R(s,t)$ are functions defining an approximation of the value function behind the boundaries and $s_{(i-,\theta)}$, s_i, $s_{(i+,\theta)}$ are states that the discretized process may attain with the probabilities $p_{(i-,\theta)}$, p_i, $p_{(i+,\theta)}$

under the control θ after the time-step Δt if the previous state was s_i. It holds $s_{(i-,\theta)} < s_i < s_{(i+,\theta)}$. In order to match the moments of this discretized process with the original time-continuous process (2.2) the probabilities are chosen in the following manner:

$$p_{(i-,\theta)} = \frac{-\mu\Delta t(\Delta_+ s - \mu\Delta t) + Var}{\Delta_- s(\Delta_- s + \Delta_+ s)}, \tag{3.3}$$

$$p_{(i,\theta)} = \frac{(-\Delta_- s - \mu\Delta t)(\Delta_+ s - \mu\Delta t) + Var}{-\Delta_- s\Delta_+ s}, \tag{3.4}$$

$$p_{(i+,\theta)} = \frac{(-\Delta_- s - \mu\Delta t)(-\mu\Delta t) + Var}{(\Delta_+ s + \Delta_- s)\Delta_+ s}. \tag{3.5}$$

Here, $\Delta_+ s = s_{(i+,\theta)} - s_i$, $\Delta_- s = s_i - s_{(i-,\theta)}$, $\mu := \mu(s_i, t_j, \theta)$ and $Var := Var(s_i, t_j, \theta)$ is chosen in such manner, that $Var/\Delta t$ is equal or at least converges to $\sigma^2(s_i, t_j, \theta)$ with $\Delta t, \Delta s \to 0$. As explained in [4], these probabilities sum up to one. However, we need to choose states $s_{(i-,\theta)}$, $s_{(i+,\theta)}$ such that all probabilities are positive. Let us assume that the drift μ is positive. Then $p_{(i+,\theta)}$ is positive, and $p_{(i-,\theta)}$, $p_{(i,\theta)}$ are positive if the following condition holds:

$$\Delta_- s\Delta_+ s + \mu\Delta t(\Delta_+ s - \Delta_- s) \geq (\mu\Delta t)^2 + Var \geq \mu\Delta t\Delta_+ s \tag{3.6}$$

We choose

$$s_{(i-,\theta)} = \left\lfloor s_i - \sqrt{(\mu(s_i, t_j, \theta)\Delta t)^2 + Var(s_i, t_j, \theta)} \right\rfloor_s, \tag{3.7}$$

$$s_{(i+,\theta)} = \left\lceil s_i + \sqrt{(\mu(s_i, t_j, \theta)\Delta t)^2 + Var(s_i, t_j, \theta)} \right\rceil_s, \tag{3.8}$$

where $\lceil\rceil_s$ denotes rounding to the nearest greater element from $s_1, s_2, \ldots, s_N$, and $\lfloor\rfloor_s$ denotes rounding to the nearest smaller element from $s_1, s_2, \ldots, s_N$. If such element does not exist, $\lceil x\rceil_s$ and $\lfloor x\rfloor_s$ will return just x. This corresponds to the boundary cases where $x < s_1$ or $x > s_N$. Now it holds

$$\sqrt{(\mu\Delta t)^2 + Var} \leq \Delta_- s, \Delta_+ s \leq \sqrt{(\mu\Delta t)^2 + Var} + \Delta s \tag{3.9}$$

and the first inequality in (3.6) holds. For the second inequality in (3.6) it is sufficient if

$$(\mu\Delta t)^2 + Var \geq \left(\sqrt{(\mu\Delta t)^2 + Var} + \Delta s\right)\mu\Delta t \tag{3.10}$$

For $Var = A(s_i, t_j, \theta)$ with

$$A(s_i, t_j, \theta) = 1/2\left(-(\mu\Delta t)^2 + 2|\mu|\Delta t\Delta s + |\mu|\Delta t\sqrt{(\mu\Delta t)^2 + 4|\mu|\Delta t\Delta s}\right) \tag{3.11}$$

condition (3.10) is fulfilled as equality, for larger Var as inequality. Therefore we set

$$Var = \max\left(\sigma^2(s_i, t_j, \theta)\Delta t, A(s_i, t_j, \theta)\right) \tag{3.12}$$

and compute $s_{(i-,\theta)}$, $s_{(i+,\theta)}$ according to (3.7), (3.8) using this value. We should note, that in (3.11) we replaced μ with $|\mu|$ to cover also the analogous case of a negative drift μ. Now, also the second part of the inequality (3.6), is fulfilled. It holds $Var/\Delta t \to \sigma^2(s_i, t_j, \theta)$ with $\Delta t, \Delta s \to 0$ and it is easy to check that the difference $|Var - \sigma^2(s_i, t_j, \theta)\Delta t|$ is smaller or equal than in the original paper [4]. Following [4], the scheme is then consistent and formula (3.12) is even better then the original version [4], as potentially less artificial diffusion is added.

4. Modification: control-independent stencil. The dependence of the possible states $s_{(i-,\theta)}$, $s_{(i+,\theta)}$ on the control variable θ implies also a dependence of $v^{j+1}_{(i-,\theta)}$, $v^{j+1}_{(i+,\theta)}$ on θ and makes the optimization problem in (3.1) harder to solve. Therefore, our goal now is to find a θ-independent choice of possible states s_{i-}, s_{i+}, while preserving condition (3.6) (and its analogue for negative drift). We will assume a positive drift $\mu(s_i, t_j, \theta)$, the case of negative drift is treated analogously.

Let us define

$$\begin{aligned} W_M &= \max_{\theta\in\Theta} \left(\sigma^2(s_i, t_j, \theta)\Delta t + (\mu(s_i, t_j, \theta)\Delta t)^2\right) \\ &= \sigma^2(s_i, t_j, \theta_M)\Delta t + (\mu(s_i, t_j, \theta_M)\Delta t)^2, \qquad (4.1)\\ E &= \max_{\theta\in\Theta} |\mu(s_i, t_j, \theta)\Delta t|\,, \qquad (4.2)\\ W_E &= 1/2\left(E^2 + 2\Delta s E + E\sqrt{E^2 + 4\Delta s E}\right). \qquad (4.3)\end{aligned}$$

It holds $W_E = E(\sqrt{W_E} + \Delta s)$ and for all $W \geq W_E : W > E(\sqrt{W} + \Delta s)$. Finally, let us define

$$W = \max(W_E, W_M) \qquad (4.4)$$

and

$$\begin{aligned} s_{i-} &= \left\lfloor s_i - \sqrt{W}\right\rfloor_s \geq s_i - (\sqrt{W} + \Delta s), \qquad (4.5)\\ s_{i+} &= \left\lceil s_i + \sqrt{W}\right\rceil_s \leq s_i + (\sqrt{W} + \Delta s). \qquad (4.6)\end{aligned}$$

Moreover, we redefine also the variance $Var(s_i, t_j, \theta)$:

$$Var = \max\left(\sigma^2\Delta t, \quad |\mu\Delta t|\,(\sqrt{W} + \Delta s) - (\mu\Delta t)^2\right), \qquad (4.7)$$

where $\sigma = \sigma(s_i, t_j, \theta)$, $\mu = \mu(s_i, t_j, \theta)$. It is easy to check that $Var/\Delta t \to \sigma^2$ as $\Delta t, \Delta s \to 0$ and therefore the consistency is preserved. Now it holds

$$\Delta_- s,\ \ \Delta_+ s \geq \sqrt{W} \geq \sqrt{W_M} = \sqrt{\sigma^2(s_i, t_j, \theta_M)\Delta t + (\mu(s_i, t_j, \theta_M)\Delta t)^2}.$$

Therefore it also holds

$$\begin{aligned} \Delta_- s\Delta_+ s + \mu\Delta t(\Delta_- s - \Delta_+ s) &\geq \sigma^2(s_i, t_j, \theta_M)\Delta t + (\mu(s_i, t_j, \theta_M)\Delta t)^2 \\ &\geq \sigma^2(s_i, t_j, \theta)\Delta t + (\mu(s_i, t_j, \theta)\Delta t)^2. \qquad (4.8)\end{aligned}$$

It also holds

$$\Delta_- s\Delta_+ s + \mu\Delta t(\Delta_- s - \Delta_+ s) \geq W \geq E(\sqrt{W} + \Delta s) \geq |\mu\Delta t|(\sqrt{W} + \Delta s). \qquad (4.9)$$

From (4.8) and (4.9) the first inequality of (3.6) holds. The second inequality of (3.6) holds, because

$$Var + (\mu(s_i, t_j, \theta)\Delta t)^2 \geq \mu \Delta t \Delta_+ s. \tag{4.10}$$

Equation (4.10) also holds if we replace $\mu \Delta t \Delta_+ s$ with $|\mu \Delta t| \Delta_- s$ which is important for the case of a negative drift. Now substituting $s_{(i-,\theta)}, s_{(i+,\theta)}$ with s_{i-}, s_{i+} for all values of θ, we get also θ-independent values $v^{j+1}_{(i-,\theta)}, v^{j+1}_{(i+,\theta)}$ (that can be written as v^{j+1}_{i-}, v^{j+1}_{i+}, and the scheme (3.1) still remains consistent and monotone ($p_{(i-,\theta)}, p_i, p_{(i-,\theta)} \geq 0$). In the next section, we employ this *"modified scheme"* to effectively solve the control problem arising in each node in equation (2.3).

5. Analytical solution of the control problem in the modified scheme. According to [4] where also relationship of the Tree-Grid method with the FDMs is discussed, the numerical scheme (3.1) can be written as

$$\begin{aligned} v_i^j &= \max_{\theta \in \Theta} \Big(f_i^j(\theta)\Delta t + (1 + r_i^j(\theta)\Delta t) \\ &\quad \cdot \Big(v_i^{j+1} + \mu_i^j(\theta)\Delta_j t D_1 v_i^{j+1} + 1/2 \Big(Var_i^j(\theta) + (\mu_i^j(\theta)\Delta_j t)^2 \Big) D_2 v_i^{j+1} \Big) \Big) \\ &:= \max_{\theta \in \Theta} F_i^j(\theta), \end{aligned} \tag{5.1}$$

where $\mu_i^j(\theta) = \mu(s_i, t_j, \theta)$, $Var_i^j(\theta) = Var(s_i, t_j, \theta)$ and D_1, D_2 are standard finite difference approximations of the first and second derivative on nonuniform grids:

$$D_1 v_i^{j+1} = \left(\frac{s_{i+} - s_i}{s_{i+} - s_{i-}} \right) \frac{v_i^{j+1} - v_{i-}^{j+1}}{s_i - s_{i-}} + \left(\frac{s_i - s_{i-}}{s_{i+} - s_{i-}} \right) \frac{v_{i+}^{j+1} - v_i^{j+1}}{s_{i+} - s_{i-}}, \tag{5.2}$$

$$D_2 v_i^{j+1} = \left(\frac{v_{i+}^{j+1} - v_i^{j+1}}{s_{i+} - s_{i-}} - \frac{v_i^{j+1} - v_{i-}^{j+1}}{s_i - s_{i-}} \right) \Big/ \left(\frac{s_{i+} - s_{i-}}{2} \right). \tag{5.3}$$

Now, under the modification presented in the previous section, s_{i+} and s_{i-} are control-independent and hence also $D_1 v_i^{j+1}$ and $D_2 v_i^{j+1}$ are control independent. Then, for a fixed node (s_i, t_j) the function $F_i^j(\theta)$ is some combination of the functions $f_i^j(\theta)$, $r_i^j(\theta)$, $\mu_i^j(\theta)$ and $Var_i^j(\theta)$. As these functions are typically in closed form, it should be possible to search for the $\max_{\theta \in \Theta} F_i^j(\theta)$ analytically, and it is not necessary to discretize Θ (if it is for example an interval).

However, $Var_i^j(\theta)$ is defined as the maximum of two different functions in (4.7) and therefore may switch its form in several points of the interval Θ. This can make the analytical computation of $\max_{\theta \in \Theta} F_i^j(\theta)$ quite difficult. This problem is not present, if we can assure $Var_i^j(\theta) = \sigma(s_i, t_j, \theta)^2 \Delta t$. That condition is typically fulfilled for a relatively large diffusion coefficient σ compared to the drift coefficient μ.

6. Fibonacci algorithm for finding the optimal control. Because of the possible complications arising by the search for the analytical solution of the control problem $\max_{\theta \in \Theta} F_i^j(\theta)$ presented in the previous section, our aim is now to present another, more straightforward approach.

Let us suppose:

1. Θ is a one-dimensional interval.
2. Discount rate $r_i^j(\theta)$ is constant in θ.
3. Increment rate $f_i^j(\theta)$ and drift $\mu_i^j(\theta)$ are linear in θ.
4. Volatility $\sigma^2(s_i, t_j, \theta)$ is convex in θ.

These conditions are fulfilled in many applications. Under these conditions, it is easy to verify, that also $1/2(Var_i^j(\theta) + (\mu_i^j(\theta)\Delta_j t)^2)$ is convex. Then, $F_i^j(\theta)$ is convex or concave and therefore has at most one local (and global) extreme inside the interval Θ and has at least one extreme on the boundary. This makes the problem $\max_{\theta\in\Theta} F_i^j(\theta)$ suitable for the Fibonacci algorithm for maximum search [1]:

Discretize the interval Θ into Φ_n points $\theta_1, \theta_2, \ldots \theta_{\Phi_n}$ where Φ_n is the n-th Fibonacci number.
Set $a = 1$, $b = \Phi_n$, $c_1 = \Phi_{n-2}$, $c_2 = \Phi_{n-1}$
for $j = n-1, n-2, \ldots, 3$ **do**
  if $F_i^j(\theta_{c_1}) > F_i^j(\theta_{c_1})$ **then**
    $b := c_2$;
    $c_2 := c_1$;
    $c_1 := a - 1 + \Phi_{j-2}$;
  else
    $a := c_1$;
    $c_1 := c_2$;
    $c_2 := a - 1 + \Phi_{j-1}$;
  end
end
$\max_{\theta\in\Theta} F_i^j(\theta) \approx \max(F_i^j(\theta_a), F_i^j(\theta_{c_1}), F_i^j(\theta_{c_2}), F_i^j(\theta_b), F_i^j(\theta_1), F_i^j(\theta_{\Phi_n}))$

Algorithm 1: Fibonacci algorithm for finding the optimal control

In the last step of the algorithm we included for testing also values $F_i^j(\theta_1), F_i^j(\theta_{\Phi_n})$ for the case that the function $F_i^j(\theta)$ is convex and the maximum is on the boundary. The computational time of the Fibonacci algorithm is $\mathcal{O}(n) = \mathcal{O}(\log(\Phi_n))$ which is much better than the computational time of the brute-force search approach [4] that is $\mathcal{O}(\Phi_n)$ for Φ_n controls.

7. Numerical experiment. We will test this modified Tree-Grid method with control-independent stencil, and the Fibonacci algorithm for control search on a Passport option pricing problem. This problem is solved with implicit FDM in [6]. In [4], a "capped payoff" is used as terminal condition, and the performance of the implicit FDM and of the Tree-Grid method is compared. Here, we will use the same parameters, terminal and boundary conditions as in [4]. For convenience we repeat here briefly the problem formulation. Passport options are contracts that allow the buyer to run a trading account for a certain amount of time. After the maturity, the buyer of this contract can keep the profit, or some part of it, however the potential loss will be covered by the seller. The HJB equation for the price of a passport option is

$$\frac{\partial V}{\partial t} + \max_{|\theta|\le 1}\Big(\frac{\sigma^2}{2}(x-\theta)^2\frac{\partial^2 V}{\partial x^2} + \Big((r-r_c-\gamma)\theta-(r-r_t-\gamma)x\Big)\frac{\partial V}{\partial x} - \gamma V\Big) = 0 \tag{7.1}$$

Here, t is time, V is the option price divided by asset price S and $x = W/S$, where W is wealth accumulated on the trading account. By r, we denote the risk-free interest rate, γ is the dividend rate, r_c is the cost of carry rate, r_t is the interest rate for the trading account and σ is the volatility. The number of shares that the investor holds (control variable) is denoted by θ, and it does not have to be an integer. In this case the seller of the option requires the constraint $|\theta| \le 1$. We used the same parameter values as in [6]: $r = 0.08, \gamma = 0.03, r_c = 0.12, r_t = 0.05, \sigma = 0.2$.

Computational domain: The maturity of the option will be one year ($T = 1$), the spatial domain will be restricted to $[-3, 4]$. The grid will be uniformly spaced in time, and non-uniformly in space. On the coarsest grid, the time-step size is 0.01. At each refinement, a four-times smaller time-step is taken. Basis for the space grid is vector of nodes:

$$S_0 = [-3, -2, -1.5, -1, -0.75, -0.5, -0.375, -0.25, -0.1875, -0.125, -0.0625, 0, 0.0625, 0.125, 0.1875, 0.25, 0.375, 0.5, 0.75, 1, 1.5, 2, 3, 4] \tag{7.2}$$

On the coarsest grid, 15 another nodes are equidistantly inserted between each two neighbouring nodes of S_0. Moreover, at each refinement, a new space-node is inserted between each two neighbouring space-nodes.

Terminal and boundary conditions: As terminal condition we use the "capped" payoff:

$$V(T,x) = V_T(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } 0 < x \le 1 \\ 1 & \text{if } x > 1 \end{cases},$$

and the Dirichlet boundary conditions:

$$V(x_{min}, t) = BC_L(x_{min}) = 0, \quad V(x_{max}, t) = BC_R(x_{max}) = 1,$$
$$x_{min} = -3, \quad x_{max} = 4.$$

Results: In the Figure 7.1, we illustrate results of numerical simulations. The left figure presents a comparison of error and computational time of the original Tree-Grid method [4] with the modified Tree-Grid method with control-independent stencil for different discretizations. To compute the error, we used as a benchmark solution a solution computed on a very fine grid (having twice as much space and time nodes as the grid at the last refinement level) with an implicit FDM from [6]. In both cases, the control interval was discretized into 9 different controls, and we used brute-force search for the optimal control. We see that the modified Tree-Grid

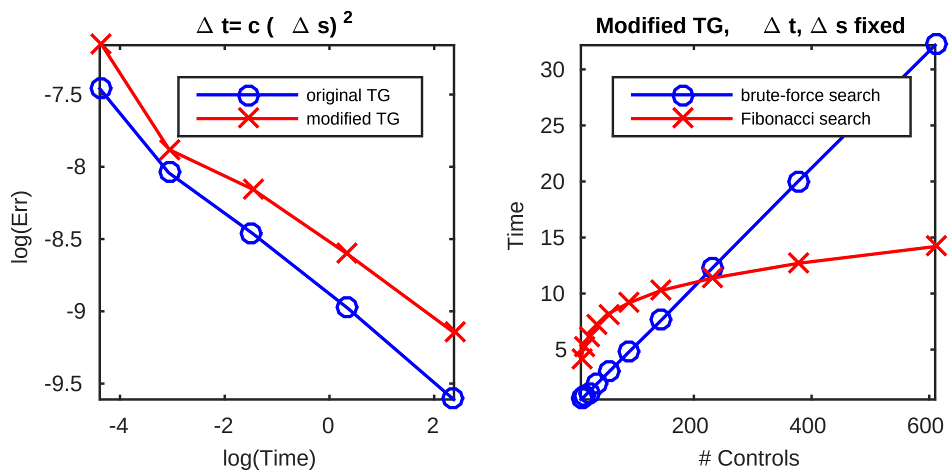

FIG. 7.1. *Left: Comparison of the natural logarithm of estimated absolute error of numerical solution against natural logarithm of computational time (in seconds) for the original Tree-Grid (TG) method and the modified Tree-Grid method with control independent stencil. Brute-force search for optimal control is done in both cases. Right: Computational time (in seconds) of the modified Tree-Grid method with control independent stencil for different number of controls in cases of brute-force search and Fibonacci search for optimal control.*

(TG) method converges, however the original method performs better. This may be of course compensated for finer discretizations of the control interval, if the optimal control is searched analytically or with a Fibonacci search algorithm in the modified scheme.

This illustrates the right figure. Here we used a coarse grid with 24 space-nodes defined by (7.2), 100 (equidistant) time-steps and a varying number of controls. As number of controls (on the x-axis), we used the Fibonacci numbers from the fifth (8) to the 14th (610). We compared the computational time of the modified Tree-Grid method with a brute-force search for control and with a Fibonacci search for control. We observe that for a large number of controls the Fibonacci search performs better due to its logarithmic time-complexity (in contrast to the linear time complexity of brute-force search). We should note that the actual values presented here in the figure are strongly implementation dependent, but they are sufficient in illustrating the proof of concept.

8. Conclusion. In this paper we presented modification of the Tree-Grid method [4] leading to a control independent “stencil” (control independent possible future states s_-^{j+1}, s_+^{j+1}). Due to this modification, it is possible to solve the optimization problem arising in each step analytically. As this approach may be still quite complicated in some cases, we proposed solving the control problem with a Fibonacci search algorithm, if certain conditions on the problem parameters are fulfilled. We analyzed the performance of the original and the modified method using an example of HJB equation from finance, and illustrated the logarithmic time-complexity of the Fibonacci search algorithm that can be applied in the modified scheme. In Section 3, we also improved the strategy of adding artificial diffusion from [4].

REFERENCES

[1] D.E. Ferguson, Fibonaccian searching. *Communications of the ACM*, 3(12):648, 1960.
[2] P. A. Forsyth, and G. Labahn, Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1, 2007.
[3] S. Kilianová, and D. Ševčovič, A transformation method for solving the Hamilton-Jacobi-Bellman equation for a constrained dynamic stochastic optimal allocation problem. *The ANZIAM Journal*, 55(01):14–38, 2013.
[4] I. Kossaczký, M. Ehrhardt, and M. Günther, A new convergent explicit Tree-Grid method for HJB equations in one space dimension. Preprint 17/06, University of Wuppertal, to appear in Numerical Mathematics: Theory, Methods and Applications, 2017.
[5] H. Kushner, and P.G. Dupuis, *Numerical methods for stochastic control problems in continuous time*, volume 24. Springer Science & Business Media, 2013.
[6] J. Wang, and P.A. Forsyth, Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 46(3):1580–1601, 2008.
[7] Jiongmin Yong, and Xun Yu Zhou, *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.