```
1: procedure threshold_search(n, r, H, B̂, B̄_n, T̄) do
2:     // Process rounds 1 and 2
3:     if ((r = 1) ∨ (r = 2)) ∧ (r ≠ n) then
4:         for all (α, β, p) in H do
5:             p_r ← p,  B̂_n ← p_1 ··· p_r B̂_{n−r}
6:             if B̂_n ≥ B̄_n then
7:                 α_r ← α,  β_r ← β,  add T̂_r ← (α_r, β_r, p_r) to T̂
8:                 call threshold_search(n, r + 1, H, B̂, B̄_n, T̂)
9:     // Process intermediate rounds
10:    if (r > 2) ∧ (r ≠ n) then
11:        α_r = α_{r−2} + β_{r−1}
12:        p_{r,min} = B̄_n/(p_1 p_2 ··· p_{r−1} B̂_{n−r})
13:        C ← ∅ // Initialize the country roads table
14:        for all β_r : (p_r(α_r → β_r) ≥ p_{r,min}) ∧ ((α_{r−1} + β_r) = γ ∈ H) do
15:            add (α_r, β_r, p_r) to C // Update country roads table
16:        for all (α, β, p) : α = α_r in H and all (α, β, p) ∈ C do
17:            p_r ← p,  B̂_n ← p_1 p_2 ... p_r B̂_{n−r}
18:            if B̂_n ≥ B̄_n then
19:                β_r ← β,  add T̂_r ← (α_r, β_r, p_r) to T̂
20:                call threshold_search(n, r + 1, H, B̂, B̄_n, T̂)
21:    // Process last round
22:    if (r = n) then
23:        α_r = α_{r−2} + β_{r−1}
24:        if (α_r in H) then
25:            (β_r, p_r) ← p_r = max_{β∈H} p(α_r → β) // Select the max. from the highway table
26:        else
27:            (β_r, p_r) ← p_r = max_β p(α_r → β) // Compute the max.
28:            if p_r ≥ p_thres then
29:                add (α_r, β_r, p_r) to H
30:        p_n ← p_r,  B̂_n ← p_1 p_2 ... p_n
31:        if B̂_n ≥ B̄_n then
32:            α_n ← α_r,  β_n ← β,  add T̂_n ← (α_n, β_n, p_n) to T̂
33:            B̄_n ← B̂_n,  T̄ ← T̂
34:    B̂_n ← B̄_n,  T̂ ← T̄ // Update the target bound and the best found trail
35:    return B̂_n, T̂
```