Blockchain Mining dengan Beberapa Penambang Egois

Qianlan Bai, Yuedong Xu, Nianyi Liu, Xin Wang

Abstrak - Makalah ini mempelajari masalah mendasar mengenai keamanan konsensus PoW blockchain tentang bagaimana keberadaan beberapa penambang yang berperilaku buruk mempengaruhi profitabilitas penambangan egois. Setiap penambang egois (atau penyerang secara bergantian) mempertahankan rantai private dan membuatnya publik secara oportunistik untuk memperoleh lebih banyak imbalan yang tidak sepadan dengan kekuatan Hash-nya. Kami pertama-tama menetapkan model rantai Markov umum untuk mengkarakterisasi transisi negara rantai publik dan swasta untuk Basic Selfish Mining (BSM), dan memperoleh *ambang* batas menguntungkan stasioner dari kekuatan Hash dalam bentuk tertutup. Ini mengurangi dari 25% untuk penyerang tunggal menjadi di bawah 21,48% untuk dua penyerang simetris secara teoritis, dan selanjutnya berkurang menjadi sekitar 10% dengan delapan penyerang simetris secara eksperimental. Kami selanjutnya mengeksplorasi ambang batas yang menguntungkan ketika salah satu penyerang melakukan penambangan strategis berdasarkan Sebagian Observable Markov Decision Process (POMDP) bahwa hanya setengah dari atribut yang berkaitan dengan keadaan pertambangan yang dapat diamati kepadanya. Algorithm online disajikan untuk menghitung kebijakan yang hampir optimal secara efisien meskipun ruang negara besar dan ruang kepercayaan dimensi tinggi. Penyerang strategis menambang dengan egois dan lebih gesit daripada penyerang BSM ketika kekuatan Hash-nya relatif tinggi, dan menambang dengan jujur sebaliknya, sehingga mengarah ke ambang menguntungkan yang jauh lebih rendah. Terakhir, kami merumuskan model sederhana dari pendapatan pertambangan absolut yang menghasilkan pengamatan yang menarik: penambangan egois tidak pernah menguntungkan pada periode penyesuaian kesulitan pertama, tetapi membalas penggantian keuntungan penambangan egois stasioner di periode mendatang. Penundaan sampai menjadi menguntungkan dari penyerang meningkat dengan penurunan kekuatan Hash-nya, membuat penambang blockchain lebih berhati-hati dalam melakukan penambangan egois.

Ketentuan Indeks—Bukti Kerja, Penambangan Egois, Profitabilitas, Rantai Markov, MDP yang Dapat Diamati Sebagian.

I. PENGANTAR

Bitcoin telah mendapatkan kekhawatiran yang luar biasa sebagai cryptocurrency pertama yang sepenuhnya terdesentralisasi sejak kemunculannya pada tahun 2008. Semua transaksi historis klien Bitcoin dicatat dalam struktur data global dan publik yang dikenal sebagai *blockchain*. Keamanan blockchain seperti Bitcoin didirikan oleh rantai teka-teki Hash kriptografi, yang ditangani oleh jaringan besar peserta pseudonim yang disebut *penambang* [1]. Memecahkan teka-teki Hash dianggap sebagai cara untuk menghasilkan Proofof-Work (PoW) untuk mencapai konsensus global. PoW Bitcoin menuntut perhitungan intensif, sehingga mengkonsumsi banyak energi. Setiap penambang bersaing untuk "permainan" ini, dan dihargai oleh mata uang crypto (yaitu bitcoin) jika dia adalah penambang pertama yang diakui untuk menemukan blok yang valid. Ketika populasi penambang besar, kekuatan Hash agregat cukup tinggi sehingga penambang jahat hampir tidak dapat mengumpulkan sumber daya yang cukup untuk melakukan pengeluaran double atau Serangan 51%. Konsensus PoW Bitcoin telah banyak digunakan di blockchain publik, berfungsi sebagai landasan cryptocurrency utama saat ini.

Q. Bai dan X. Wang bersama Sekolah Ilmu Komputer, Universitas Fudan, shanghai 200438, Cina.

Y. Xu dan N. Liu bersama Sekolah Ilmu dan Teknologi Informasi, Universitas Fudan, 200438 Shanghai, Cina.

Keamanan PoW ditantang oleh tren sentralisasi kekuatan Hash. Menambang blok Bitcoin adalah acak dan membutuhkan rata-rata lebih dari 10 tahun dengan chip ASIC generasi terbaru [4]. Oleh karena itu, penambang blockchain beroperasi secara kooperatif untuk membentuk kolam yang memiliki peluang jauh lebih besar untuk memecahkan teka-teki dalam waktu singkat. Dengan membagi hadiah penambangan dengan tepat, mereka memperoleh tingkat pendapatan yang stabil. Sebagai efek samping, sejumlah kecil kolam penambangan menempati sebagian besar kekuatan Hash global, menempatkan sistem blockchain dengan risiko digulingkan oleh kolam raksasa atau kolam berkolusi. Kebijaksanaan konvensional percaya bahwa PoW aman selama tidak ada pool pertambangan (atau penambang secara bergantian) yang mengendalikan 51% dari total kekuatan Hash. Namun, seorang penambang dapat memilih untuk menambang dengan egois daripada menyesuaikan diri dengan protokol Bitcoin standar.

Penambangan egois mengacu pada kelas kebijakan penerbitan blok di mana penambang tidak segera release blok yang baru ditemukan, tetapi garpu rantai pribadi yang tidak disadari orang lain. Pada zaman masa depan, ia akan melepaskan blok pribadinya secara strategis untuk usang rantai publik saat ini atau bersaing dengannya untuk tujuan mendapatkan blok yang lebih tinggi dalam rantai publik baru daripada rasio kekuatan Hash-nya. Singkatnya, penambang egois tidak ingin menghancurkan konsensus PoW blockchain , tetapi untuk memanfaatkannya. Rasio minimum kekuatan Hash yang membawa lebih banyak hadiah bagi penambang egois daripada rasio ini secara konvensional disebut *ambang menguntungkan*. Eyal dan Sirer memperkenalkan skema penambangan egois pertama (yaitu penambangan egois dasar, BSM) dan menunjukkan bahwa ambang batas BSM yang menguntungkan adalah 25% dari total kekuatan Hash [2]. Nayak dkk. [9] mengusulkan penambangan keras kepala yang meningkatkan pendapatan penambang egois sebesar 13,94% dibandingkan dengan BSM. Sebagai trik utama dari skema keras kepala, penambang egois bersikeras untuk forking jika rantai pribadinya sedikit tertinggal di belakang rantai publik. Sapirshtein et al. Memodelkan penambangan egois yang optimal sebagai Markov Decision Process (MDP) yang mengurangi ambang menguntungkan penambang egois menjadi 23,21% [7]. Tao et al. [20] memperkenalkan serangan penambangan semi-egois berdasarkan MDP tersembunyi dengan kontrol tingkat garpu. Grunspan dan PerezMarco terbukti secara ketat menggunakan teori martingale bahwa

penambangan egois adalah serangan terhadap algoritma penyesuaian kesulitan konsensus blockchain [32]. Baru-baru ini, banyak upaya telah dikhususkan untuk serangan majemuk penambangan egois dengan serangan menahan block [14] [34], serangan penyuapan [15], serangan gerhana dan serangan pengeluaran ganda [22].

Sampai baru-baru ini, persaingan beberapa penambang egois mulai terlihat. Liu et al. mempresentasikan skema *publish-n* untuk dua penyerang pertambangan egois dan mensimulasikan pendapatan mereka serta ambang batas yang menguntungkan [3]. Bai dkk. Memodelkan perlombaan penambangan di antara dua penambang BSM dan satu penambang jujur sebagai proses Markov[31]. Zhang et al. [18] mensimulasikan threshold yang menguntungkan di hadapan beberapa penambang egois. Charlie et al. [19] mengusulkan SquirRL, kerangka kerja untuk menggunakan pembelajaran penguatan mendalam (DRL) untuk menganalisis penambangan egois dan memblokir serangan pemotongan dalam sistem blockchain. Pengaturan multi-agen mereka secara roughly setara dengan beberapa penambang egois, dan fokus mereka adalah untuk melatih model DRL yang sangat berkinerja baik untuk mengatasi penambangan strategis dengan ruang negara yang besar dan informasi yang tidak lengkap. Studi eksperimental sebelumnya, meskipun berwawasan luas, kurangnya understandings teoritis pada prinsip penambangan egois kompetitif. Biasanya diyakini bahwa pemodelan ambang menguntungkan dengan beberapa penambang egois sulit, dan penambangan yang optimal mungkin sulit diatasi karena tantangan yang disebutkan di atas.

Dalam makalah ini, kami secara teoritis menyelidiki profitabilitas penambangan egois dengan banyak penyerang dengan mengajukan serangkaian pertanyaan progresif utama: *1) Akankah penambangan egois menjadi lebih mudah menguntungkan dengan banyak penyerang daripada dengan penyerang solo? 2) Bagaimana kita bisa merancang kebijakan penambangan yang hampir optimal untuk penyerang meskipun interaksi yang kompleks di antara penambang dan pengamatan yang tidak lengkap dari keadaan sistem? 3) Berapa lama penyerang BSM harus menunggu dari awal penambangan egois sampai akhirnya menguntungkan?* Mencari tahu pertanyaan-pertanyaan ini akan memberikan pemahaman penting tentang keamanan konsensus PoW blockchain .

Kami merumuskan model rantai Markov untuk menghitung *pendapatan relatif* stasioner penyerang BSM untuk pertanyaan pertama. Pendapatan relatif penyerang adalah persentase dari blok *yang valid* dalam rantai publik. Model kami sangat umum dalam arti bahwa ia dapat menangkap kasus dengan lebih dari dua penyerang atau memungkinkan penyerang untuk menyembunyikan beberapa blok secara pribadi. Secara khusus, kasus latter dapat menyebabkan pelepasan chainreaction yang rumit di mana tindakan penerbitan satu penyerang memicu penyerang lainnya.

Menjawab pertanyaan kedua sangat menantang jika penyerang strategis (Alice), penyerang BSM (Bob) dan penambang honest (Henry) hidup berdampingan dalam sistem. Pertama, interaksi antara tiga rantai lebih rumit. Keadaan perlombaan pertambangan ditangkap oleh 10-tuple yang berkaitan dengan komposisi semua rantai dan status forking, sebagai lawan dari 3-tuple yaitu kebijakan optimal berbasis MDP untuk penyerang tunggal. Jumlah tindakan lebih besar dan pasangan tindakan negara bisa 102 hingga 103 kali lebih besar. Kedua, Alice sebagai penambang strategis tidak dapat mengamati informasi negara yang rumit. Faktanya, dia hanya mengurus 5 atribut di setiap negara bagian yang terkait dengan rantai pribadinya dan rantai publik. Kami merumuskan keluarga model proses keputusan Markov (POMDP) yang dapat diamati sebagian parameter untuk mengkarakterisasi kebijakan penambangan strategis Alice dengan belief berkelanjutan dari keadaan saat ini. Untuk mengatasi ruang negara yang besar dan ruang kepercayaan dimensi tinggi, kami mengadopsi AEMS2, salah satu algoritma online POMDP tercepat, untuk menghitung kebijakan penambangan yang hampir optimal. Metode pencarian biner yang mirip dengan [7] digunakan untuk menemukan pendapatan relatif maksimum di antara keluarga model POMDP.

Adapun pertanyaan ketiga, kami membangun model sederhana untuk menghitung *pendapatan absolut* penyerang, yang merupakan jumlah rata-rata blok valid yang diterima oleh setiap penambang per unit waktu. Karena mining egois adalah serangan terhadap algoritma penyesuaian kesulitan (DAA), itu tidak menguntungkan secara instan bahkan jika kekuatan Hash penyerang berada di atas ambang batas yang menguntungkan. Model ini memungkinkan kita untuk menghitung jumlah periode DAA yang mengarah pada penambangan selfish yang menguntungkan pada akhirnya.

Pengamatan utama kami diringkas seperti di bawah ini.

- BSM. Ambang batas kekuatan Hash yang menguntungkan di bawah 21,48% dengan dua penyerang BSM simetris, dibandingkan dengan 25% dengan penyerang BSM tunggal dan 23,21% dengan penyerang optimal tunggal. Lebih banyak blok yang diizinkan untuk dipegang secara pribadi atau lebih banyak penyerang akan secara dramatis mengurangi ambang batas ini. Ketika kekuatan Hash dari dua penyerang asimetris, ambang menguntungkan dari satu penyerang akan berkurang terlebih dahulu dan kemudian meningkat ketika power Hash penyerang lainnya meningkat (yaitu tidak monoton).
- POMDP. Kebijakan pertambangan POMDP membawa lebih banyak pendapatan bagi penambang strategis daripada BSM dan pertambangan jujur , dan mendekati kinerja kebijakan penambangan MDP dengan informasi yang lengkap. Ketika penyerang BSM (Bob) memiliki kekuatan Hash 34%, ambang menguntungkan penyerang lain (Alice) menurun dari 29,44% menjadi sekitar 2% jika dia memilih POMDP daripada BSM. Algoritma online yang dirancang dapat dengan cepat dan efektif menghitung

tindakan yang hampir optimal di bawah information yang dapat diamati saat ini.

- Penundaan yang menguntungkan. Seorang penambang BSM menerima pendapatan yang kurang absolut daripada penambangan jujur pada periode penyesuaian kesulitan pertama bahkan jika kekuatan Hash-nya berada di atas ambang batas yang menguntungkan, dan keuntungannya dicapai di periode mendatang. BSM adalah after yang menguntungkan 51 putaran penyesuaian kesulitan (yaitu 714 hari dalam Bitcoin) jika kekuatan Hash dari dua penambang egois simetris adalah 22%. Penundaan ini menurun menjadi 5 putaran (yaitu 70 hari dalam Bitcoin) karena kekuatan Hash mereka bertambah menjadi 33%, yang masih cukup lama.

Sisa-sisa makalah ini disusun sebagai berikut. Bagian II menggambarkan latar belakang penambangan egois. Bagian III memodelkan pendapatan relatif dari penambangan egois dasar dengan penyerang yang berbeda. Bagian IV mengusulkan kebijakan pertambangan berbasis POMDP dan merancang algoritma eficient. Waktu yang menguntungkan dari penambangan egois dasar dimodelkan dalam Bagian V. Bagian VI memvalidasi model pendapatan BSM dan kinerja kebijakan berbasis POMDP. Bagian VII memperkenalkan pekerjaan terkait, dan Bagian VIII menyimpulkan pekerjaan kami.

II. SYSTEM MODEL

Pada bagian ini, kami menyajikan prosedur pelepasan blok penambangan blockchain di hadapan dua penambang yang bermusuhan. Kami selanjutnya memperkenalkan fitur-fitur baru pada tie-breaking dan rilis reaksi berantai.

A. Deskripsi Sistem

Pertimbangkan sistem blockchain dengan dua penyerang nakal *Alice* dan *Bob*, serta penambang yang jujur, *Henry*[1]. Mereka bersaing untuk memecahkan teka-teki kriptografi untuk menambang blok yang valid untuk tujuan memperoleh token seperti bitcoin. Konsensus proof-of-work (PoW) diadopsi dan penambangan blok tidak memiliki kewarganegaraan: kemungkinan menemukan blok oleh penambang bersifat proportional terhadap kekuatan Hash-nya saat ini, tetapi berbanding terbalik dengan kekuatan Hash agregat saat ini dari seluruh jaringan blockchain. Sistem blockchain secara dinamis menyesuaikan kesulitan teka-teki kriptografi sehingga blok baru dihasilkan pada tingkat rata-rata tetap (misalnya, satu blok per 10 menit rata-rata dalam Bitcoin). Para penambang mempertahankan serangkaian transaksi yang dipesan secara global melalui adopsi dan penambangan pada rantai terpanjang. Pendapatan relatif penambang adalah sebagian kecil dari blok yang ditambang olehnya dari semua blok dalam rantai terpanjang. Hadiah dari setiap blok yang valid dinormalisasi sebagai satu koin kriptografi.

Untuk kesederhanaan, kami membuat asumsi berikut konsisten dengan literatur [2] [7]. Lingkungan penambangan blockchain memiliki that

- Total kekuatan Hash dari sistem blockchain dinormalisasi sebagai satu unit. Kemudian, kekuatan Hash penambang direpresentasikan sebagai sebagian kecil dari total;
- Waktu penemuan blok oleh penambang didistribusikan secara eksponensial.

Penambang jujur Henry yang menemukan blok yang valid akan segera melepaskannya. Alice (resp. Bob) dapat melepaskan bloknya secara strategis dengan memaksa Henry membuang-buang perhitungannya. Ketika Alice dan Bob sama-sama penambang egois, interaksi antara dua rantai pribadi menjadi lebih rumit karena tidak ada dari mereka yang tahu keadaan orang lain. Berikut ini, kami menangkap semua negara bagian yang berbeda yang mungkin dihadapi setiap penambang.

Menunjukkan dengan α_1, α_2 dan α_h kekuatan Hash Alice, Bob dan Henry masing-masing, yaitu, $\alpha_1 + \alpha_2 + \alpha_h = 1$. Tunjukkan dengan γ_1 (resp. γ_2) proporsi yang ditambang oleh Henry's Hash setelah alice 's (resp. Bob's) melepaskan rantai dalam tie-breaking antara Alice (resp. Bob) dan Henry. Tunjukkan dengan $\vartheta 1$ dan θ_2 probabilitas bahwa penambang jujur memilih untuk menambang setelah rantai Alice dan Bob di tie-breaking tiga pihak, masing-masing. Ketika sistem blockchain membuat blok baru, itu ditambang oleh pool i dengan probabilitas α_i, $\forall i \in \{1,2,h\}$, karena memori perhitungan Hash.

B. Mode Penambangan Egois Dasar

Alice mempertahankan rantai pribadi, begitu juga Bob, sementara Henry beroperasi pada rantai publik. Alice dan Bob tidak menyadari peran masing-masing, bahkan kehadiran satu sama lain. Kami menduga bahwa semua penambang bekerja pada rantai publik yang sama di awal di mana titik awal dinyatakan sebagai "0". Panjang rantai pribadi disimpan sebagai informasi pribadi oleh Alice dan Bob, dan panjang rantai publik diamati oleh mereka semua. Kami consider metode penambangan egois yang diusulkan oleh [2], dan pendekatan analitis kami dapat digeneralisasikan ke berbagai metode lain.

Prosedur penambangan terdiri dari dua kasus sebagai berikut.

- *(Kasus penambangan rantai publik)* Henry selalu menambang setelah rantai publik. Alice atau Bob juga menambang di rantai publik jika lebih panjang dari rantai pribadinya.

[1] Beberapa penambang yang jujur dapat direbus menjadi penambang tunggal demi additivity linier kekuatan Hash mereka.

- *(Kasus penambangan rantai swasta)* Alice (resp. Bob) terus menambang rantai pribadinya (resp. nya) jika dia (resp. dia) menemukan blok baru dan rantai pribadi sekarang lebih panjang than rantai publik.

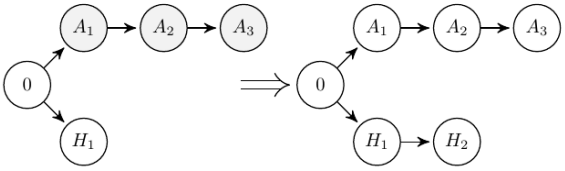
Prosedur pelepasan lebih rumit daripada prosedur penambangan. Henry menyiarkan blok yang ditambang segera setelah ditemukan, sementara Alice dan Bob akan memutuskan apakah akan melepaskan blok yang ditambang tergantung pada panjang rantai publik.

- *(Kasus hangus)* Alice (resp. Bob) meninggalkan rantai pribadinya (resp. nya) dan sesuai dengan penambangan setelah rantai publik jika yang terakhir lebih panjang. Henry juga meninggalkan rantai publiknya jika Alice atau Bob menerbitkan rantai yang lebih panjang.
- *(Kasus pelepasan yang menghindari risiko)* Alice (resp. Bob) melepaskan bloknya (resp. nya) yang ditambang secara pribadi ke publik karena takut kehilangan jika blok baru ditambang oleh yang lain dan keuntungan utama dari rantai pribadinya tidak lebih dari dua blok.
- *(Kasus reaksi berantai)* Ketika Alice (resp. Bob) melepaskan bloknya (resp. nya) ke rantai publik dan memperbarui panjangnya, pelepasan blok pribadi Bob (resp. Alice) dipicu segera.

Kasus reaksi berantai adalah kombinasi dari kasus-kasus yang hilang dan menghindari risiko, sedangkan keberadaan reaksi berantai mempersulit evolusi rantai publik. Misalkan Alice menerbitkan blok pribadinya untuk usang rantai publik saat ini. Setelah pembangunan rantai publik baru, Bob mungkin release rantai pribadinya untuk kehilangan itu segera.

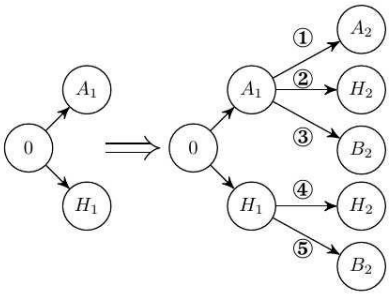
C. Prosedur pelepasan dan Logika tie-breaking

Konsensus tentang rantai publik mensyaratkan bahwa itu adalah yang terpanjang. Pertanyaan penting adalah bagaimana rantai publik berkembang ketika memiliki panjang yang sama dengan Alice atau Bob. Secara umum, setiap penambang bekerja pada rantainya sendiri, dan perilaku pelepasan Alice dan Bob dipicu ketika Henry menambang blok baru. Kami dengan ini menggambarkan evolusi rantai pribadi dan publik di mana *Ak*, *Bk*, dan *Hk* menunjukkan bahwa blok *kth* milik Alice, Bob dan Henry masing-masing. Blok rantai pribadi berwarna abu-abu dan rantai publik berwarna putih. Kasus pelepasan yang menghindari risiko. Kami menunjukkan pelepasan rantai pribadi Alice yang menghindari risiko di Gambar 1. Alice hanya satu blok di depan Henry setelah yang terakhir menambang blok baru untuk rantai publik. Karena Alice takut kehilangan kompetisi, dia menerbitkan blok pribadinya, mengaburkan rantai publik Henry, sehingga Alice dan Henry menambang di rantai terpanjang baru sesudahnya.



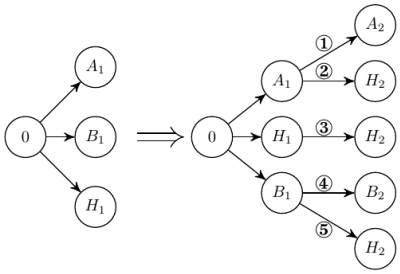
Gambar 1. Pelepasan alice yang menghindari risiko dan pengabaian Henry.

Tie-breaking menyelesaikan. Jika rantai pribadi Alice hanya satu blok di depan Henry, Henry mungkin menyusulnya. Ketika itu terjadi, Alice segera menerbitkan blok pribadinya untuk bersaing dengan Henry. Dengan demikian, dua rantai publik dengan panjang yang sama ada di Gambar 2. Karena hanya satu rantai publik yang berlaku, aturan tie-breaking perlu diperhitungkan. Kasus pertama adalah bahwa rantai publik Alice dan Henry memiliki panjang yang sama, dan rantai private Bob adalah 0. Oleh karena itu, kita hanya perlu menyelesaikan ikatan antara Alice dan Henry. Semua penambang dimungkinkan untuk menambang setelah blok *A1*, sementara Bob dan Henry dapat menambang setelah *H1*. Ada lima kemungkinan untuk memperluas rantai publik terpanjang, dan yang lebih pendek akan usang. Kami menghilangkan tie-breaking antara Bob dan Henry karena ini dapat dianalisis dengan cara yang sama.

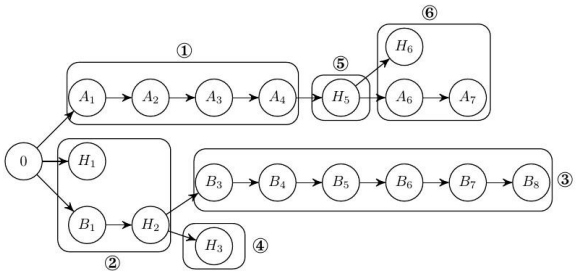


Gambar 2. Kasus tie-breaking dari dua rantai publik.

Untuk situasi bahwa masing-masing Alice dan Bob menyembunyikan satu blok pribadi, mereka akan mempublikasikan rantai pribadi their langsung setelah Henry menemukan blok baru. Seperti yang ditunjukkan pada Gambar 3, ada tiga rantai publik yang bersaing. Alice akan menambang setelah $_{A1}$ dan Bob akan menambang setelah *B1* pasti; Henry tidak menyadari rantai mana yang bercabang dengan jahat sehingga ia dapat menambang di setiap rantai publik. Ada juga lima kemungkinan situasi. Pelepasan penghindaran risiko, bersama dengan dua solusi tie-breaking, merupakan semua dinamika rantai pribadi dan publik.



Gambar 3. Kasus tie-breaking dari tiga rantai publik.



Gambar 4. Kasus reaksi berantai.

Pelepasan reaksi berantai. Kami selanjutnya memperkenalkan rilis reaksi berantai yang mempersulit evolusi rantai pribadi dan publik. Perhatikan bahwa pelepasan reaksi berantai terdiri dari urutan pelepasan penghindaran risiko dan penyelesaian tie-breaking.

Gambar 4 menggambarkan contoh bagaimana fenomena reaksi berantai dipicu. Pada tahap 1, rantai pribadi Alice berisi empat blok, sedangkan panjang rantai pribadi Bob dan rantai publik Henry adalah 0. Setelah resolving tie-breaking pada tahap 2, rantai publik yang lebih panjang berisi dua blok *B1* dan $_{H2}$, dan yang lebih pendek yatim piatu. Bob membangun rantai pribadi baru mulai dari *B3* hingga $_{B8}$, sementara Henry terus menambang satu blok setelah *H2* pada tahap 4. Dari sudut pandang Alice, rantai private-nya hanya satu blok di depan rantai publik. Dia melepaskan blok pribadinya untuk menghindari risiko kalah dalam perlombaan dengan Henry. Rantai publik baru sekarang dimulai dari blok *A4*. Selanjutnya, tahap 5 dan 6 merupakan babak baru resolving tie-breaking antara Alice dan Henry, memperluas rantai publik untuk memblokir *A7*. Namun, rilis *A7* memicu Bob untuk merilis semua blok pribadinya mulai dari *B3* hingga *B8*. Ketika retrospeksi semua tahap penambangan, kami mengamati bahwa cabang yang menang beralih bolak-balik, membuat analisis penambangan egois sangat rumit. Perlu dicatat, reaksi berantai terjadi hanya ketika panjang rantai pribadi lebih besar dari tiga.

III. ANALISIS STRIKASI DARI BASIC SELFISH MINING

Di bagian ini, kami menyajikan model rantai Markov untuk mengkarakterisasi dinamika penerbitan blok dengan beberapa penambang egois. Pendapatan yang diharapkan dari penambang egois berasal dalam bentuk eksplisit.

A. Definisi

Tema penelitian kami ditempatkan pada profitabilitas penambangan egois sehingga langkah-langkah yang menguntungkan harus diklarifikasi terlebih dahulu. Untuk kesederhanaan notasi, kami hanya mempertimbangkan tiga penambang: Alice, Bob dan Henry.

Definisi 1: (*Pendapatan Relatif*) *Biarkan Ra, Rb dan Rh menjadi jumlah yang diharapkan dari blok yang valid yang ditambang oleh Alice, Bob dan Henry di putaran penambangan, masing-masing. Pendapatan relatif seorang penambang,* $\hat{R}_i$*, dinyatakan sebagai*

$$\hat{R}_i = \frac{R_i}{R_a + R_b + R_h}, \qquad i \in \{a, b, h\}$$

.

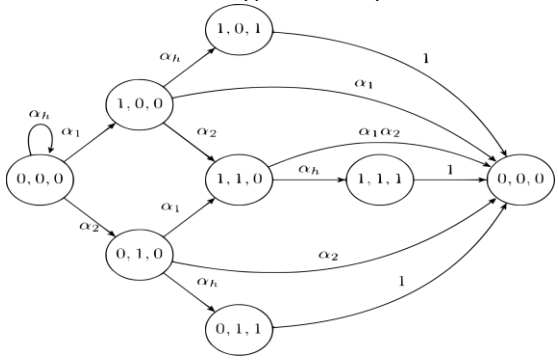
Perlu ditekankan bahwa blok yang valid adalah blok yang dikonfirmasi dalam rantai longest. Profitabilitas penambangan egois tidak mengacu pada surplus bahwa hadiah blok mengurangi biaya perhitungan kriptografi. Bahkan, ini adalah ukuran kontras dengan penambangan jujur yang membutuhkan indeks objektif.

Definisi 2: (*Profitability*) *Penambangan egois atau strategis yang dilakukan oleh Alice (resp. Bob) dianggap menguntungkan jika pendapatan relatif lebih tinggi dari kekuatan Hash yang dinormalisasi, yaitu* $\hat{R}_a > \alpha_1$ *(resp.* $\hat{R}_b > \alpha_2$).

B. Analisis Stasioner untuk Dua Penyerang

Untuk menganalisis profitabilitas penambangan egois, kita perlu menghitung pecahan blok yang ditambang oleh penambang egois dalam rantai publik. Kami dengan ini merumuskan model rantai Markov diskrit untuk mengkarakterisasi dinamika rantai publik dan pribadi. Kita mulai dengan asumsi bahwa setiap penambang egois akan melepaskan dua blok pribadinya segera setelah ia telah menambang yang kedua (yaitu *N* = 2).

Alasan yang mendasarinya adalah dua kali lipat. Pertama, proses pelepasan blok yang lebih sederhana menghindari representasi rumit dari keadaan Markov, sehingga memungkinkan pemodelan matematika yang lebih mudah diselok. Kedua, pelepasan bursty dari banyak blok yang valid dalam waktu yang sangat singkat biasanya menunjukkan adanya serangan penambangan egois yang dapat dengan mudah dideteksi. Para penambang egois berniat untuk memperoleh imbalan penambangan tambahan selain mendestabilisasi otoritas cryptocurrency.



Gambar 5. Rantai Markov dengan kurang dari dua blok private.

Kami mendefinisikan *keadaan* sebagai tiga tuple yang terdiri dari panjang rantai Alice, Bob dan Henry. Gambar 5

menggambarkan semua negara bagian, indikator transisi negara dan probabilitas transisinya. Misalnya, transisi dari negara bagian (*0,0,0*) ke negara bagian (*1,0,0*) berarti bahwa Alice menemukan blok yang valid dengan probabilitas α_1 dan garpu rantai pribadi. Jika panjang maksimum rantai pribadi di bawah 2, Alice dan Bob dapat menyembunyikan rantai pribadi mereka dan melanjutkan penambangan egois. Semua transitions untuk menyatakan (0,0,0) berarti bahwa rantai bercabang kembali ke rantai publik bulat dan babak baru pertambangan egois dimulai. Menunjukkan dengan **P** matriks probabilitas transisi keadaan dan dengan P_{ss0} probabilitas transisi dari keadaan $s = (i, j, k)$ ke $s0 = (i0, j0, k0)$. Biarkan π_{ijk} menjadi distribusi stasioner negara (i,j,k). Menurut *persamaan keseimbangan terperinci* [5] [21]

$$\pi_{s0} = \sum_{s} \pi_{s} P_{s s0}. \quad (1)$$

Kemudian, kita dapat menghitung $p000$ sebagai $p000 = (1 + \alpha_1 + \alpha_2 + \alpha_1\alpha_h + \alpha_1\alpha_2 + \alpha_2\alpha_h + \alpha_1\alpha_2\alpha_h)^{-1}$, (2)

dan π_{ijk} pada keadaan lain $s = (i, j, k)$ dengan cara yang sama.

Transisi ke negara bagian (0,0,0) memanifestasikan penambang mana yang merupakan pemenang akhir dalam putaran penambangan egois saat ini. Oleh karena itu, kita dapat menghitung pendapatan yang diharapkan dari Alice, Bob dan Henry yang didefinisikan sebagai R_a, R_b dan R_h masing-masing. Difasilitasi oleh distribusi status stasioner, kami menghitungnya seperti di bawah ini,

$$R_a = \pi_{000} \cdot [2\alpha_1^2(1+\alpha_h) + (\alpha_2+\alpha_h)\alpha_1\alpha_h\gamma_1 + \alpha_1\alpha_2\alpha_h + 4\alpha_2^2\alpha_1(1+\alpha_h) + 2\alpha_1\alpha_2\alpha_h^2\theta_2]; \quad (3)$$

$$R_b = p000 \cdot [2\alpha_2^2(1+\alpha_h) + (\alpha_1+\alpha_h)\alpha_h 2c_2 + \alpha_1\alpha_2\alpha_h + 4\alpha_2^2\alpha_1(1+\alpha_h) + 2\alpha_1\alpha_2\alpha_h^2\theta_2]; \quad (4)$$

$$R_h = p000 \cdot [\alpha_1\alpha_h 2(2-c_1) + 2\alpha_1\alpha_2\alpha_h 2\ (2-i_1-\vartheta_2) + \alpha_h + \alpha_2\alpha_h 2(2-c_2) + \alpha_1\alpha_2\alpha_h(2-c_1-c_2)]. \quad (5)$$

Jumlah rata-rata blok yatim piatu Henry di setiap putaran serangan dihitung sebagai:

$$O_h = p000[(\alpha_1+(1-\alpha_1)c_1)_{a1ah} + (\alpha_2+(1-\alpha_2)c_2)\alpha_2\alpha_h + (\alpha_1+\alpha_2+(i_1+i_2)\alpha_h)2a1a2ah]. \quad (6)$$

Sebagai kasus khusus bahwa kedua penambang egois itu homogen, yaitu $\alpha_1 = \alpha_2 = \alpha < 0.5$, $\gamma_1 = \gamma_2 = 0.5$ dan $\vartheta_1 = \vartheta_2 = 1/3$, pendapatan yang diharapkan dapat disederhanakan sebagai $\pi_{000} = (1 + 4\alpha - 4\alpha^3)^{-1}$;

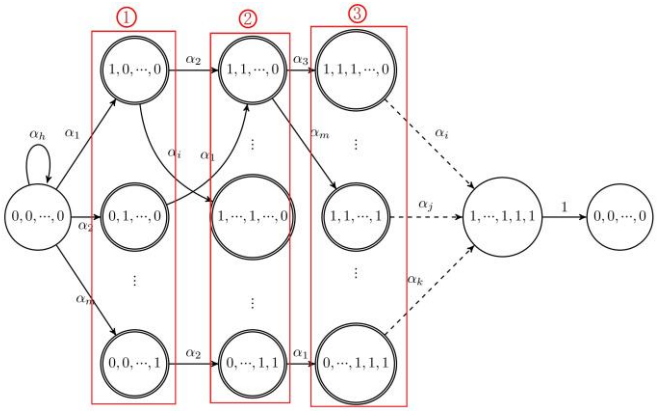
$$R_a = R_b = \pi_{000} \cdot \tfrac{1}{6}\alpha(25\alpha + 2\alpha^2 + 3 - 32\alpha^3); \quad (7)$$

$$R_h = \pi_{000} \cdot [(1-2\alpha)(1+3\alpha - \tfrac{7}{3}\alpha^2 - \tfrac{16}{3}\alpha^3)]. \quad (8)$$

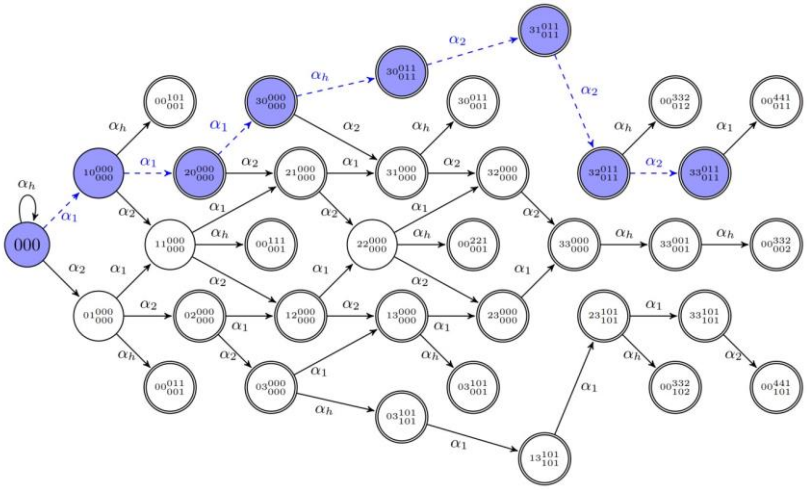
Kita dapat dengan mudah mengamati bahwa pendapatan yang diharapkan penyerang (resp. Henry) di Eq. (7) (resp. Eq. (8)) secara monoton meningkat (resp. penurunan) sehubungan dengan rasio penyerang dari kekuatan Hash.

C. Penskalaan ke Beberapa Penyerang

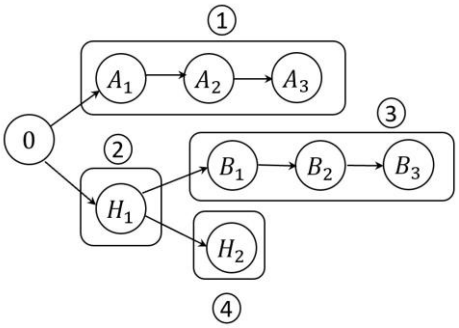
Meskipun penambangan egois yang menguntungkan menuntut kekuatan Hash yang tinggi, ada kemungkinan bahwa beberapa penambang egois dapat memilih untuk ikut serta. Dampak penambang yang lebih egois terhadap profitabilitas tidak jelas. Bagian penambang yang jujur dari kekuatan Hash menurun, dan persaingan di antara penambang egois menjadi lebih sengit ketika jumlah penambang egois meningkat lebih dari dua. Oleh karena itu, kami mempertimbangkan skenario mire umum dengan *penambang BSM m* ($m > 2$).



Gambar 6. Transisi Negara Markov dengan Penyerang *m* .



Gambar 7. Mesin negara dengan $N = 4$.



Gambar 8. Sampel jalur dengan $N = 4$.

Kami memodelkan dinamika rantai publik dan pribadi sebagai proses Markov juga. Gambar 6 menggambarkan semua keadaan dan probabilitas transisinya. Ingatlah bahwa asumsi panjang rantai pribadi maksimum juga berlaku. Setiap keadaan dinyatakan sebagai m-tuple, yaitu $L = \{l_1, l_2, \cdots, l_m\}$ dengan $l_i \in \{0,1\}$, yang terdiri dari panjang rantai pribadi masing-masing penyerang. Keadaan dengan *m* nol, dilambangkan sebagai *L0*, menunjukkan awal dari kompetisi mining. Negara-negara dengan satu '1' dikelompokkan bersama di mana satu dan hanya satu penyerang telah menambang blok yang valid untuk membangun rantai pribadinya. Demikian pula, negara-negara dengan elemen *k* '1' menunjukkan bahwa rantai pribadi *penyerang k* out of *m* memiliki satu blok yang valid. Secara formal, kami menunjukkan dengan L set semua negara dengan kardinalitas 2m, dan oleh Lk ⊆ L subset di mana *k* penambang egois memegang blok pribadi mereka.

Probabilitas transisi negara digambarkan sebagai berikut. Sistem blockchain dapat berada di keadaan awal dengan probabilitas α_h, yaitu, blok ditambang oleh penambang yang jujur. Tunjukkan *dengan ei* vektor yang elemen *ith-nya* adalah 1 dan yang lainnya adalah zero. Ketika blok yang valid ditemukan oleh penyerang *i*, sistem blockchain transit ke keadaan baru (*L* + *ei*) dengan probabilitas α_i. Di negara bagian *L* ∈ Lk, jika penyerang *ith* yang memiliki blok pribadi menemukan blok yang valid lagi, sistem blockchain kembali ke keadaan awal *L0* (di mana negara dinyatakan sebagai lingkaran ganda). Jika tidak, ia melompat ke keadaan di L_{k+1} dengan probabilitas setara dengan kekuatan Hash relatif dari penambang egois. Probabilitas transisi negara dapat dinyatakan sebagai:

$P_{LL0} = \alpha_i \ \forall L0 = L + e_i \in L_{k+1}$ dan $L \in L_k$,

$P_{L0L0} = \alpha_h$,

$P_{LL0} = \alpha_i + \alpha_h, P_{L0L0} = 1. \forall L \in L_k, L0 \in L/_{k+1}$.

Dengan menggunakan persamaan keseimbangan terperinci, kita dapat menghitung distribusi stasioner dari setiap keadaan. Secara khusus, probabilitas stasioner di negara *S0* dihitung secara eksplisit sebagai

$$\pi_{L_0} = \left(1 + \sum_{k=1}^{m} \sum_{L \in \mathcal{L}_k} k! \prod_{j=1}^{m} (\alpha_i \cdot 1_{l_j=1})\right)^{-1} . \quad (9)$$

Probabilitas stasioner pada keadaan *L* diberikan oleh:

$$\pi_L = k! \cdot \pi_{L_0} \cdot \prod_{j=1}^{m} (\alpha_i \cdot 1_{l_j=1}), \forall L \in \mathcal{L}_k . \quad (10)$$

Pendapatan semua penambang dihitung berdasarkan distribusi keadaan stasioner dan jalur transisi tertentu ke negara *L0*. Jika semua penambang memiliki γ yang sama dalam kasus tie-breaking, pendapatan mereka dapat ditulis sebagai:

$$R_i = \sum^{m} \sum \begin{cases} \pi_L \cdot [\alpha_h (2\alpha_i + \frac{1 - \sum_{j \in L} \alpha_j \cdot 1_{l_j=1}}{k+1})], & l_i = 1; \\ \sum_{k=1} \sum_{L \in L_k} \alpha_h \cdot \pi_L \cdot \alpha & i, l_i = 0. \end{cases}$$

$$R_h = \left(\sum_{k=1}^{m} \sum_{L \in \mathcal{L}_k} \alpha_h \pi_L \alpha_h \left(\frac{2}{k+1} + \frac{k}{k+1}\right)\right) + \pi_{L_0} \cdot \alpha_h. \quad (11)$$

D. Penskalaan ke N ≥ 2

Kami selanjutnya memodelkan pendapatan semua penambang ketika setiap rantai pribadi dapat menyembunyikan lebih dari satu blok. Terutama, karena jumlah maksimum blok pribadi untuk satu penyerang tidak kurang dari empat (yaitu *N* = 4), "reaksi berantai" terjadi dan mesin keadaan terbatas resulting menjadi sangat rumit. Suatu negara harus mencakup tidak hanya panjang semua rantai, tetapi juga interleaving blok pada mereka. Kami membatasi studi kami untuk tiga penambang: Alice, Bob dan Henry, dan menyelidiki kasus *N* = 4 without kehilangan keumuman. Kesulitan utama yang menghambat analisis matematika adalah bahwa Alice dan Bob memiliki keyakinan yang berbeda dalam posisi awal putaran penambangan saat ini. Selain itu, blok dalam rantai pemenang mungkin milik Henry dan Alice / Bob sehingga kita perlu menghafalnya untuk menghitung pendapatan mereka. Sebaliknya, baik Alice dan Bob selalu memiliki posisi awal yang sama dalam balap tanpa reaksi berantai (yaitu keadaan (*0,0,0*) di Fig.5).

Transisi status dengan *N* = 4 dinyatakan dalam Gambar 7 di mana setiap keadaan terdiri dari delapan parameter. Putaran penambangan saat ini dimulai pada node paling kiri di mana tidak ada penambang yang menemukan blok. Notasi *h1* menunjukkan jarak antara posisi awal yang diyakini oleh Alice dan posisi starting yang sebenarnya. Demikian pula, *h2* dan *h3* menunjukkan jarak Bob dan Henry ini. Kami mencatat h1, *h2* dan *h3* karena blok antara posisi awal yang nyata dan yang diyakini mempengaruhi rantai mana yang akan menang akhirnya dan bagaimana pendapatan dihitung. Notasi *l1* (resp. *l2*) mewakili jumlah blok yang belum dirilis di chain pribadi Alice (resp. Bob). μ_1 menunjukkan jumlah blok Henry antara posisi awal yang sebenarnya dan posisi awal yang diyakini Alice. μ_2 dan μ_3 didefinisikan untuk Bob dan Henry dengan cara yang sama. Menggabungkan mereka bersama-sama, kami mendefinisikan keadaan Markov karena $l_1 l_2{}^{h_1 h_2 h_3}_{\mu_1 \mu_2 \mu_3}$ itu juga berlaku untuk situasi dengan *N* > 4.

Dengan ini kami menyajikan contoh konkret dari transisi negara.

Keadaan terkait dalam Gambar 7 ditandai dengan warna biru, dan transisi mereka diilustrasikan dalam Gambar 8 secara terpisah. Pada tahap 1, Alice telah menambang tiga blok secara diam-diam sehingga keadaan sistem melompat dari 0 hingga 300000000 hingga dan 10^{000}_{000} 200000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000 0000000 Pada tahap 2, Henry menambang blok *H1* yang valid dan segera menerbitkannya ke rantai publik. Keadaan sistem kemudian melompat dari 300000000 ke 30011011. Bob telah

menambang tiga blok setelah *H1* pada tahap 3 dan keadaan sistem pindah ke 33011011. Sejauh ini, baik Alice maupun Bob tidak akan melepaskan blok pribadi mereka. Pada tahap 4, Henry menemukan blok H2 baru yang memicu aksi rilis Alice. Setelah Alice menerbitkan semua bloknya ke rantai Henry yang usang, Bob menemukan bahwa rantai publik sedang mengejar ketinggalan. Akibatnya, Bob menerbitkan semua bloknya dan memenangkan kompetisi akhirnya, yaitu keadaan sistem kembali ke posisi menyatakan. Di babak ini, Bob menerima tiga hadiah blok dan Henry receives hadiah satu blok.

Menurut probabilitas transitif, pendapatan untuk setiap penambang dapat direpresentasikan sebagai:

$$p_{000} = 1/(1 + a1 + a2 + a1a3 + a_{12} + 2a2a1 + a_{22} + a_{13} + a_{2a3} + a_{13} + 3a_{2a12} + 2a1a2a3 + 3a1a22 + a_{23} + a_{13a3} + 4a_{13a2} + 6a_{12a22} + 4a_{1a23} + a_{23a3} + a_{13a2a3} + 10a13a22 + 6a12a22a3 + 10a12a23 + 4a1a23a3a3 + a_{1a23a3} + a_{13a22a3} + 20a13a23 + a_{13a23a3} + a_{13a22a32} + 20a13a23a3a3 + 4a_{13a2a3} + a_{14a23a3} + 20a13a23a32 + a_{12a23a3} + a_{13a23a3} + a_{12a23a32} + \alpha_1^3\alpha_2^4\alpha_3); \quad (12)$$

$$R_{p1=000} \cdot [4\,a14\,(1 + a_h) + 3a13ah2 + 16a14a2 + 4a12ah + 40a14a22\,(1 + 2a2) + a_{1a2ah}\,(1 + c_1 + 2i1ah) + 10a12a2ah + 20a13a2ah\,(3a2 + a1) + 15a13a2ah2 + 4a14a22ah\,(1 + ah) + 4a14a23ah2\,(b1 + 20) + 5a15a23ah + 4a14a23ah\,(a2 + 21) + 3a13a24ah2b1 + a_{1ah2c1} + 12a12a_{22ah2b1} + a12a22ah3b1\,(3a1 + 2a2) + 6\alpha_1^3\alpha_2^3\alpha_h^2\,(10\alpha_h\beta_1 + 1)]; \quad (13)$$

$$R_2 = p_{000}[4a24\,(1 + a_h) + 3a23ah2 + 16a1a24 + 4a22ah + 40a12a24\,(1 + 2a1) + a1a2ah\,(1 + c2 + 2i2ah) + 10a1a22ah + 20a1a23ah\,(3a1 + a2) + 15a1a23ah2 + 4a12a24ah\,(1 + ah) + 4a13a24ah2\,(b2 + 20) + 5a13a25ah + 4a13a24ah\,(a1 + 21) + 3a14a23ah2b2 + a_{2ah2c2} + 12a12a_{22ah2b2} + a12a22ah3b2\,(2a1 + 3a2) + 6\alpha_1^3\alpha_2^3\alpha_h^2\,(10\alpha_h\beta_2 + 1)]; \quad (14)$$

$$R_h = p_{000}[a1ah2\,(2 - c1) + a2ah2\,(2 - c2) + a_{12a23ah3}\,(2b1 + b2) + 2a1a2ah2\,(2 - \vartheta 1 - \vartheta 2) + a_{12a22ah2}\,(6 + 4a1a2) + a_{13a22ah3}\,(b1 + 2\,b2) + a1a2ah\,(2 - c_1 - c2) + a_{13a23ah}\,(a_1 + a2) + a_{13a24ah2}\,(2b1 + b_2) + ah + a_{14a23ah2}\,(b1 + 2b2) + 20a13a23ah3 + 2\alpha 14a24ah]; \quad (15)$$

$$b1 = c1/(c_1 + c2)\ b2 = c2/(c1 + c2). \quad (16)$$

Kasus dengan $N > 4$ dapat dianalisis dengan cara yang sama. Kami memvalidasi dalam eksperimen kami bahwa pendapatan relatif cenderung menyatu ketika $N \geq 4$. Jika N terlalu besar, reaksi berantai berulang akan terjadi, yang memperburuk ketidakstabilan sistem dan meningkatkan kemungkinan detecting serangan pertambangan egois.

IV. STRATEGI OPTIMAL DI BAWAH BEBERAPA PENYERANG

Kebijakan pertambangan egois dasar membatasi pilihan menahan dan melepaskan blok. Pada bagian ini, kami menyajikan strategi penambangan egois yang *optimal* (*policy pertambangan berbasis POMDP*) untuk dua penyerang ketika salah satu dari mereka memilih penambangan egois dasar dan yang lainnya memilih untuk menjadi strategis.

A. MEMILIH *Batas Atas Pendapatan*

Keterbatasan penambangan egois dasar intuitif. Seorang penyerang "konservatif" untuk mengadopsi chain publik ketika rantai pribadinya sedikit tertinggal di belakangnya , dan "kurang bijaksana" untuk mengesampingkan rantai publik ketika mengejar ketinggalan. Masalah penambangan egois yang optimal dengan penyerang tunggal diangkat pada [7] [22] yang memodelkan perlombaan penambangan sebagai process keputusan Markov. Di bawah kebijakan optimal, penyerang tidak mengadopsi rantai publik jika rantai pribadinya sedikit lebih pendek dari rantai publik; dia mungkin sengaja melepaskan beberapa blok untuk menyebabkan forking jika rantai pribadinya berada di depan rantai publik oleh beberapa blok. Kebijakan penambangan egois strategis ini menurunkan ambang kekuasaan Hash yang menguntungkan.

Penambangan egois yang optimal (OPT) di hadapan dua penyerang (Alice sebagai *penyerang strategis* dan Bob sebagai *penyerang dasar*) jauh lebih menantang dengan penyerang tunggal. Pertama, negara dan transisi negara sangat ditambah. Alice harus menggabungkan status beberapa rantai dalam keadaan selain hanya panjang rantai. Balap tiga rantai juga menyebabkan transisi keadaan yang terlibat. Kedua, Alice tidak dapat memperoleh informasi mengenai rantai pribadi Bob. Dengan kata lain, keadaan sebagian dapat diamati oleh Alice. Untuk mengatasi kesulitan ini, kita mulai dengan asumsi bahwa Alice memiliki informasi lengkap tentang rantai pribadi Bob. Kebijakan optimal Alice dapat diselesaikan berdasarkan model MDP, dan pendapatan yang sesuai akan digunakan

sebagai batas atas pendapatan ketika rantai pribadi Bob tidak diketahui. Sementara itu, model MDP menawarkan prinsip of merancang kebijakan optimal dengan keadaan yang dapat diamati sebagian.

1) Komponen utama: Kami merumuskan model MDP untuk penyerang strategis sebagai M empat tuple = < S, A, P, R > di mana S menunjukkan ruang negara, A menunjukkan ruang aksi, P sesuai dengan matriks transisi, dan R sesuai dengan matriks hadiah.

Status: Ruang negara S didefinisikan sebagai 10-tuple dalam bentuk < *loc, fork, l1, l2, h1, h2, h3, u1, u2, u3* >. Atribut *loc* ∈ {*1,2,3*} menunjukkan cabang yang sedang dikerjakan Henry. Jika *loc* = 1 (resp. *loc* = 2), Henry menambang pada rantai publik yang juga berisi blok Alice (resp. Bob' di putaran penambangan saat ini. Jika *loc* = 3, rantai publik terpanjang hanya berisi blok Henry. Perhatikan bahwa blok Alice dan Bob saling eksklusif pada rantai publik di putaran penambangan yang sama karena satu penyerang tidak akan menerima blok yang lain sebelum putaran serangan ini berakhir.

Garpu *atribut* memperoleh enam nilai, dijuluki sebagai {*ir*, *r*, $_{f12}$, $_{f13}$, $_{f23}$, $_{f123}$}, di mana r mewakili bahwa Alice dapat melepaskan blok untuk bersaing dengan rantai publik saat ini ketika *h3* > 0 sementara *Ir* mewakili bahwa dia tidak bisa. Jika beberapa penambang bersaing di rantai publik, *garpu* mengambil empat nilai {$_{f12,f13}$, $_{f23}$, $_{f123}$}, menunjukkan bahwa (Alice, Bob), (Alice, Henry), (Bob, Henry) dan (Alice, Bob, Henry) masing-masing berada dalam kompetisi. Atribut lain seperti *l1*, *l2*, *h1*, *h2*, *h3*, μ_1, μ_2 dan μ_3 memiliki arti yang sama dengan yang ada di rantai Markov tersebut. Similar untuk [7], kami membatasi panjang rantai pribadi dan publik dalam putaran penambangan sehingga membatasi ukuran ruang negara, yaitu *l1* ≤ *N* dan *h3* ≤ $_{h3,max}$. Tindakan: Tindakan adalah jumlah blok yang diterbitkan Alice di bawah keadaan tertentu. Kami mendefinisikan ruang aksi Alice A sebagai A = {*adopt,0,1,*· ·· ,$_{l1}$} di mana *mengadopsi* berarti bahwa Alice menyerahkan rantai pribadinya, 0 berarti bahwa Alice memilih untuk *menunggu*, dan *l1* adalah jumlah blok saat ini yang dipegang oleh Alice secara pribadi. Tindakan yang diambil oleh Alice memiliki batasan yang wajar: jika *l1* mencapai *N*, Alice harus melepaskan setidaknya satu blok; jika *h3* mencapai $_{h3,max}$, Alice memilih "mengadopsi" atau melepaskan tidak kurang dari (*h3−h1*) blok untuk mengakhiri penambangan ini bulat.

Transisi Status: Kami mendefinisikan fungsi transisi status sebagai Pr(*s0*| *s,* a ∈ A), probabilitas bahwa negara melompat ke negara ^{s0} di bawah tindakan *a*. Transisi negara dipicu oleh penambangan blok baru, dan ditentukan oleh penambang mana yang menemukannya. Semua transisi diringkas dalam Tabel III.

Fungsi Hadiah: Tujuan penambangan "optimal" adalah untuk memperoleh bagian yang lebih besar dari blok yang dikonfirmasi pada rantai publik, atau untuk mengerucutkan pendapatan relatif yang lebih besar dengan kata lain. Ingatlah bahwa pendapatan relatif seorang penambang adalah sebagian kecil dari bloknya pada rantai publik untuk periode long. Jelas, pendapatan relatif tidak dapat diukur di bawah setiap pasangan tindakan negara, dan tidak dapat diambil sebagai hadiah langsung yang sesuai. Sapirshtein et al. [7] mengubah pendapatan relatif (jangka panjang) menjadi keluarga (satu tembakan) pendapatan absolut yang diparameterisasi oleh berat *ρ* ∈ [*0,1*].

Transformasi cerdik ini beroperasi sebagai berikut. Tentukan fungsi transformasi *wρ* : N3 → R terkait dengan

Hadiah instan Alice:

$$w_\rho^i(r_1^i, r_2^i, r_h^i) = (1-\rho)\cdot r_1^i - \rho\cdot(r_2^i + r_h^i), \quad (17)$$

di mana r_1^i, r_2^i dan $_{rhi}$ mewakili imbalan seketika Alice, Bob dan Henry pada langkah *i* (analog dengan waktu *t* dalam MDP klasik). Kami merumuskan kembali model MDP asli sebagai $_{Mρ}$ = < S, A, P, *wρ* (*r1, r2,* $_{rh}$) >. Alasan yang mendasari transformasi tersebut adalah bahwa alih-alih memaksimalkan pendapatan relatif, kami memilih untuk memaksimalkan hadiah fiktif yang diharapkan *wρ* (*r1, r2,* $_{rh}$). Untuk setiap π kebijakan yang dapat diterima, hadiah rata-rata yang dilambangkan v_ρ^π ditandai sebagai:

$$v_\rho^\pi = \mathbb{E}[\varliminf_{\xi\to\infty} \frac{1}{\xi}\sum_{i=1}^{\xi} w_\rho(r_1^i(\pi), r_2^i(\pi), r_h^i(\pi))], \quad (18)$$

di mana *ξ* adalah jumlah total langkah transisi negara. Pendapatan optimal *vρ** diberikan oleh

$$v_\rho^* = \max_\pi \{v_\rho^\pi\}. \quad (19)$$

Kesetaraan antara dua mdps M dan Mρ tidak langsung. Sapirshtein et al. [7] menyajikan dua proposisi untuk menjamin kesetaraan mereka untuk penambang egois tunggal.

- If $v_\rho^* = 0$ untuk beberapa *ρ* ∈ [*0,1*], maka kebijakan apa pun *π** memperoleh nilai ini juga memaksimalkan pendapatan relatif, dan pendapatan relatif sama dengan *ρ*.
- v_ρ^* secara monoton menurun dalam *ρ*.

Proposisi di atas memberi tahu kita bahwa dengan mencari *ρ* yang sesuai yang menghasilkan hadiah rata-rata Mρ menjadi 0, kita dapat memperoleh pendapatan relatif yang optimal dari Alice di M. Kami menggeneralisasi ide ini ke MDP dengan beberapa penambang selfish. Selain itu, jumlah maksimum langkah, *ξ*, dipotong untuk menghindari perhitungan yang berlebihan dengan menoleransi kerugian yang sangat lembut dalam hadiah rata-rata optimal v_ρ^π.

2) Algoritma: Karena monotonitas *vρ** *hingga ρ*, pencarian biner *ρ* ∈ [*0,1*] sudah memadai. Untuk *ρ* tertentu, kami menggunakan metode *iterasi nilai* untuk menyelesaikan kebijakan optimal $\pi_\rho{}^*$ sebagai [7]. Dibandingkan dengan *iterasi kebijakan*, keuntungan dari *iterasi nilai* adalah tingkat konvergensi yang cepat terutama di anggota parlemen skala besar [12] [13].

B. Kebijakan berbasis POMDP untuk Pertambangan Optimal

Kerangka KERJA OPT memberikan wawasan penting tentang kebijakan penambangan yang optimal di hadapan dua penyerang, namun penyebaran dunia nyatanya tidak realistis. Penambang strategis Alice diasumsikan tahu persis keadaan sistem, sementara pada kenyataannya rantai pribadi Bob tidak dapat diamati oleh Alice, dan blok pada rantai publik yang dirilis oleh Bob dan Henry tidak dapat dibedakan karena anonimitas mereka. Mengingat informasi negara yang tidak lengkap, kami merumuskan kembali penambangan optimal sebagai Proses Keputusan Markov yang Dapat Diamati Sebagian (POMDP). Sebelum menyelami detailnya, kami menyebutkan tiga tantangan utama:

- bagian informasi mana yang tidak dapat diamati oleh Alice;
- bagaimana model MDP berbasis peristiwa penambangan di generalized ke model POMDP;
- bagaimana kebijakan pertambangan optimal berbasis POMDP dapat dihitung secara efisien.

1) Komponen utama: Model POMDP dinyatakan sebagai MPO enam tuple :=< *S,A,P,R,O,Z* >, di mana O adalah ruang observasi Alice, Z(·) adalah fungsi pengamatan, dan komponen yang tersisa mewarisi arti yang sama dengan rekan-rekan mereka di MDP.

Informasi yang Dapat Diamati: Dalam lingkungan yang dapat diamati sebagian, Alice tidak dapat memperoleh semua atribut di S. Panjang chain *l2* pribadi Bob tidak dapat diamati sepenuhnya. Atribut *loc* tidak dapat diamati baik karena Alice tidak menyadari apakah Henry menambang pada rantainya sendiri atau rantai Bob. *h2*, μ_2 dan μ_3 tidak dapat diamati karena anonimitas penambangan menutupi pemilik blok the pada rantai publik. Garpu atribut dapat diamati karena *ir* dan *r* berkaitan dengan rantai pribadi Alice, dan nilai $f_* \in$ { f12,f13,f23, $_{f123}$} diperoleh dengan menghitung jumlah focks dalam kompetisi. *l1*, *h1* dan μ_1 diketahui dengan pasti; *h3* sebenarnya adalah panjang rantai publik terpanjang. Singkatnya, ruang observasi diwakili sebagai O :=< *fork,l1,h1,h3,*μ_1 >⊂ S. Mengingat pengamatan yang sama, Alice kemungkinan berada di banyak keadaan yang mungkin.

Transisi Status: Fungsi transisi status juga dilambangkan sebagai Pr(*s0*| *s, a* ∈ *A*). Meskipun mengambil bentuk yang sama, MPO memiliki logika transisi keadaan yang berbeda dari M. Di M, tindakan an dipicu oleh penemuan blok baru, dan transisi negara mengikuti. Di $_{MPO}$, transisi negara *murni yang digerakkan oleh acara* akan membatasi Alice untuk berpartisipasi dalam kompetisi garpu. Misalnya, jika Bob menyembunyikan satu blok, Alice harus publish blok pribadinya sementara tidak ada peristiwa yang dapat diamati untuk memicu kompetisi garpu. Sebaliknya, jika Bob tidak memiliki blok pribadi sementara Alice percaya bahwa panjang rantai pribadi Bob adalah 1, Alice dapat menerbitkan satu atau dua blok yang tidak perlu. Oleh karena itu, orang dapat melihat bahwa model POMDP yang dibatasi waktu plus event-driven sesuai untuk menangani masalah penambangan optimal yang rumit.

Karena perhitungan Hash tanpa memori, proses kedatangan blok sebenarnya adalah proses stokastik stasioner. If kami mengiris proses stochastic ini sama dengan durasi slot Δ*t*, jumlah blok yang ditambang di setiap slot memiliki distribusi yang sama. Dengan memilih Δ*t* yang relatif kecil, kami menganggap bahwa paling banyak satu blok ditambang di setiap slot (kemungkinan menambang dua blok atau lebih lebih lebih jarang dengan urutan besarnya). Tunjukkan dengan *p* probabilitas menghasilkan blok dalam satu slot waktu. Probabilitas bahwa Alice, Bob dan Henry menghasilkannya adalah α_{1p}, α_{2p} dan α_{hp} masing-masing. Alice dapat memperkirakan kekuatan Hash α_2 dan α_{jam} melalui penambangan dengan jujur untuk periode tertentu. Perlu digarisba diketahui bahwa model POMDP kami membuat penambangan optimal dengan status yang dapat diamati sebagian layak, dan sesuai dengan sistem blockchain yang realistis. Semua transisi diringkas dalam Tabel IV.

Fungsi pengamatan: Define Z:= S × A → Δ(O) sebagai fungsi pengamatan yang menentukan hubungan antara keadaan sistem dan pengamatan. Di sini, *z* (*s, a, o*) adalah probabilitas bahwa pengamatan *o* akan dicapai setelah Alice melakukan tindakan *a* dan mendarat di negara bagian *s*:

$$_{zt+1}(s,a,o) = \Pr(ot_{+1} = o \mid st_{+1} = s, at = a). \quad (20)$$

Dalam MPO, pengamatan suatu negara adalah pasti,

yaitu, *s* = < *loc, garpu,* $_{l1}$*, l2, h1, h2, h3,* μ *1.*μ *2.*μ 3 >, *o* = < *garpu,* $_{l1}$*,* $_{h1}$*,* $_{h3}$ *,* μ_1 >,

$$_{zt+1}(s,a,o) = 1. \quad (21)$$

Keyakinan: Model POMDP kami berkaitan dengan keyakinan *b* yang merupakan distribusi probabilitas atas semua keadaan yang mungkin. Secara intuitif, Alice menebak keadaan saat ini secara berulang. Keyakinan pada keadaan tertentu *pada* waktu *t* diberikan oleh: *b* (*s*) = Pr (*st* = *s*| *zt,at−1,zt−1,*⋯ $_{,a0,b0}$). (22)

Keadaan kepercayaan yang diperbarui *b0* (*s0*) dihitung setiap kali tindakan *a* diambil dan pengamatan *o* dirasakan.

$$_{b0}(s0) \equiv \Pr(s0 \mid a,o,b) = \frac{\Pr(s0,a,o,b)}{\Pr(a,o,b)} \quad (23)$$

$$= \frac{\cancel{\Pr(o \mid s0,a)} \sum_{s} \Pr(s0 \mid a,s)\cancel{b(s)}}{\Pr(o \mid a,b)}, \quad (24)$$

di mana $\sum_{s\in S} b(s) = 1$ dan Pr(*o*| *a,b*) adalah faktor normalisasi yang diberikan oleh

$$\Pr(o \mid a,b) = \sum_{s0s} \Pr(o \mid a,s0)\Pr(s0 \mid a,s)b(s). \quad (25)$$

Fungsi hadiah: Kami menggunakan $r_1^i(s,a), r_2^i(s,a)$ dan $_{rhi}$(*s, a*) untuk menunjukkan hadiah instan Alice, Bob dan Henry pada langkah *i* ketika Alice mengambil tindakan *di* negara bagian *s*. Karena ketidakpastian keadaan sistem, fungsi penghargaan yang diharapkan $r_1^i(b,a), r_2^i(b,a)$ dan $r_h^i(b,a)$ dibangun di atas keyakinan
hadiah instan, ∀*a* ∈ A,

$$r_{1i}(b,a) = Xbi(s)_{r1i}(s,a), \quad (26)$$

$$r_{2i}(b,a) = Xbi(s)_{r2i}(s,a), \quad (27)$$

$$r_h^i(b,a) = \sum_{s\in S} b_i(s) r_h^i(s,a). \quad (28)$$

Tujuan Alice adalah untuk memaksimalkan hadiah relatifnya yang dapat diubah menjadi ekspresi alternatif dari hadiah absolutnya di setiap slot waktu. Mirip dengan transformasi dalam model MDP, kami mengganti hadiah relatif dengan hadiah absolut yang di $w_\rho^i(r_1^i(b_i,a), r_2^i(b_i,a), r_h^i(b_i,a))$ parameter oleh *ρ* menggunakan Eq. (17).

Kami selanjutnya merumuskan penambangan optimal sebagai masalah POMDP hadiah rata-rata cakrawala terbatas sebagai [7] dan [22]. Fungsi nilai rata-rata yang diharapkan didefinisikan sebagai

$$v_\rho^\pi = \mathbb{E}[\lim_{\xi\to\infty} \frac{1}{\xi} \sum_{i=1}^{\xi} w_\rho(r_1^i(b_i,\pi), r_2^i(b_i,\pi), r_h^i(b_i,\pi))].$$

Kebijakan optimal π^* adalah seperangkat aturan keputusan tergantung pada pasangan belief-state:

$$p^* = \text{argmax}_{\pi\in\mathcal{A}} \{v_\rho^\pi\}. \quad (29)$$

Parameter *ρ* yang dipecahkan $v_\rho^* = 0$ adalah pendapatan relatif Alice di bawah model POMDP. Dalam praktiknya, jumlah pendapatan lebih dari *ξ* dipotong oleh jumlah yang cukup besar *ξ0*. Mengingat ambang presisiϵ, ξ_0 perlu satisfy:

$$|v_\rho^\pi - \mathbb{E}[\frac{1}{\xi_0} \sum_{i=1}^{\xi_0} w_\rho(r_1^i(\pi), r_2^i(\pi), r_h^i(\pi))]| \le \epsilon. \quad (30)$$

C. Algoritma

POMDP pada dasarnya adalah MDP yang diperluas yang didefinisikan pada ruang kepercayaan, dan metode iterasi nilai klasik dapat diadopsi untuk menyelesaikan kebijakan optimal secara offline. Namun, ruang kepercayaan adalah ruang kontinu dimensi tinggi yang perlu tersegmentasi menjadi sejumlah negara kepercayaan. Algoritma POMDP offline akan menghitung tindakan optimal di setiap keadaan kepercayaan. Mempertimbangkan POMDP skala besar seperti kita, perhitungan offline memakan waktu karena menghasilkan imbalan, memperbarui keyakinan dan constructing kebijakan optimal pada setiap keyakinan. Hal ini berbeda dengan MDP yang hanya memiliki negara kepercayaan yang tepat. Untuk pertimbangan efisiensi, kami mengusulkan menggunakan algoritma POMDP online yang mengeksplorasi negara-negara kepercayaan masa depan yang dapat dijangkau dari negara belief saat ini. Waktu konstruksi kebijakan seringkali jauh lebih pendek. Selain itu, tiga properti penting dapat digunakan untuk mengurangi waktu mencari *ρ*.

Lemma 1: Di bawah pengaturan parameter yang sama, hasil optimal M *adalah batas atas* $_{MPO}$*.*

Bukti: Di antara algoritma perkiraan masalah POMDP, pendekatan MDP terdiri dari mendekati fungsi nilai POMDP dengan fungsi nilai MDP yang mendasarinya [36]. Fungsi nilai ini adalah batas atas pada fungsi nilai POMDP [28].

Lemma 2: Pendapatan yang diperoleh berdasarkan kebijakan optimal berdasarkan ρ adalah batas bawah dari pendapatan optimal aktual untuk MPO*.*

Bukti: Dalam algoritma online, langkah-langkah konstruksi kebijakan dan langkah-langkah eksekusi saling terkait satu sama lain. Oleh karena itu, kami mencatat jumlah blok yang diperoleh setiap penambang di setiap langkah waktu, yaitu (r_1^i, r_2^i, r_h^i). Setelah langkah-langkah waktu yang cukup, kita dapat menghitung pendapatan relatif di bawah ρ saat ini meskipun tidak optimal. Oleh karena itu, pendapatan dari kebijakan optimal saat ini yang diperoleh di bawah ρ saat ini adalah batas bawah dari pendapatan optimal aktual.

Lemma 3: Ada kebijakan deterministik stasioner yang optimal untuk model MPO*.*

Bukti: States, tindakan dan keyakinan dapat dihitung karena panjang maksimum rantai pribadi dan publik terbatas. Karena masalah POMDP dapat dianggap sebagai MDP kepercayaan, keberadaan kebijakan stasioner yang optimal untuk masalah POMDP kami dijamin oleh Theorem 7.3.6 dari [10].

Algoritma penambangan online kami dijelaskan dalam Algoritma 1. Kami menghitung pendapatan optimal berdasarkan pencarian biner. Batas atas dapat diatur sebagai hasil dari model MDP yang mendasarinya sesuai dengan Lemma 1 (baris 2-3). Alice akan melakukan tindakan optimal berdasarkan ρ saat ini dan memperoleh pendapatan relatif (baris 16 dan 24). Pendapatan optimalnya tidak kurang dari pendapatan relatif menurut Lemma 2 (baris 25-29). Kita tidak perlu menghitung ulang tindakan optimal di setiap langkah sesuai dengan Lemma 3. Algoritma online akan mengadopsi tindakan yang sesuai jika keadaan kepercayaan yang sama telah bertemu. Jika tidak, itu akan menghitung dan menyimpan tindakan optimal baru (baris 11-15). Sebuah blok ukuran 1MB membutuhkan 18 detik untuk mencapai tiga ribu node di Bitcoin. Membuat keputusan tepat waktu oleh Alice sangat penting. Pada baris 5, kami menggunakan AEMS2 [11] [28], salah satu algoritma POMDP tercepat, untuk menghitung tindakan optimal (baris 14).

V. ANALISIS PROFIABILITAS YANG TIDAK BERSUSAHTIK

Pada bagian ini, pertama-tama kami menjelaskan algoritma penyesuaian difficulty (DAA) dalam sistem seperti Bitcoin , dan memodelkan pendapatan penambang dalam satu periode

penyesuaian kesulitan. Kami secara analitis menunjukkan bahwa pendapatan tambahan dari penambangan egois berasal dari DAA.

A. Penyesuaian Kesulitan seperti Bitcoin

Inti dari penambangan Bitcoin adalah untuk memecahkan teka-teki kriptografi. Header blok terutama mencakup Hash dari blok sebelumnya, Hash root Merkle transaksi, waktu awal menghitung hash header, nBits yang digunakan untuk ~~menghasilkan kesulitan target dan *NONCE*. Algoritma Penambang Bitcoin 1 Algoritma untuk memecahkan POMDP.~~

Input: MPO*,* M, parameter pemotongan ξ_0, nilai presisi ε;
Keluaran:*r*;
Statis: *bc*: Keyakinan saat ini;
a^* : tindakan optimal;

1: ← *rendah* 0;
2: ρ^* =*QMDP(*M*)*;
3: ← *atas* ρ^*;
4: sementara *> ε atas - rendah* lakukan
5: $\rho \leftarrow$ (*rendah* + *atas*)*/*2
6: HASIL←{};
7: *r1* ← *0,r2* ← *0,rh* ← 0;
8: *vρ* ← 0;
9: *bc* ← keyakinan awal
10: sementara *ξ0 >* 0 lakukan
11: jika *bc* ∈ HASIL maka
12: a^* =HASIL[*bc*]
13: 14: $a^* \leftarrow$ *AEMS2*($_{bc,MPO}$)
15:00 akhiri jika
16: $(r_1^i, r_2^i, r_h^i) \leftarrow$ Jalankan a^* untuk $_{bc}$.
17: $r_1 += r_i^i, r_2 += r_2^i, r_h += r_h^i$;
18:
19: HASIL[*bc*] = a^*
20: Memahami pengamatan baru *z*
21: $_{bc} \leftarrow {}^{b0}(_{bc,a^*,z})$B algoritma update adalah Eq. (23)
22: $_{x0}$− = 1
23: akhiri sementara
24: $R_1' = r_1/(r_1 + r_2 + r_h)$;
25: jika *v* > 0 maka
$v_\rho += w_\rho(r_1^i, r_2^i, r_h^i)$;

^{R}
26: $low \leftarrow max(\rho, R_1')$;
27: lainnya
28:← *atas ρ*; ← *lebih rendah* max(*rendah, R10*);
29: akhiri jika
30: akhiri sementara 31: kembali *ρ*;

Berulang kali menyebutkan *NONCE* sampai hash kepala berada di bawah target kesulitan. Semakin kecil nilai target, semakin sulit penemuan *NONCE* yang valid. Untuk kesulitan target tetap, daya Hash yang lebih besar berarti waktu yang lebih singkat untuk menemukan *NONCE* yang valid.

Untuk mempertahankan interval pembangkit blok yang dapat dipertahankan, Bitcoin dan Altcoin memperkenalkan algoritma penyesuaian kesulitan (DAAs) untuk mengatasi kekuatan Hash variabel dalam sistem. Bitcoin DAA dieksekusi setelah blok 2016 ditambang. Ini sebenarnya adalah system kontrol umpan balik: jika waktu sebenarnya penambangan blok 2016 lebih besar dari 20160 menit (10 menit per blok), kesulitan target menurun secara proporsional, dan meningkat sebaliknya. Untuk menghindari fluktuasi yang berlebihan, kesulitan periode berikutnya harus berada dalam kisaran waktu $[\frac{1}{4}, 4]$dari kesulitan saat ini. Ketika seorang penambang melakukan penambangan egois, banyak blok menjadi yatim piatu sehingga waktu sebenarnya untuk menambang blok 2016 menjadi lebih lama. Pada periode penyesuaian kesulitan berikutnya, kesulitan target diturunkan untuk maintain tingkat pembangkit blok tetap.

B. Pendapatan Absolut

Sebelumnya kami mendefinisikan pendapatan penambang di setiap putaran penambangan dan pendapatan relatifnya. Namun, durasi putaran penambangan mungkin tidak diperbaiki sepanjang waktu, dan jumlah sebenarnya dari blok yang valid yang diperoleh oleh penambang di setiap unit waktu waktu waktu dinding diabaikan. Dalam sistem Bitcoin, kami menunjukkan 10 menit sebagai waktu unit, dan menunjukkan periode DAA sebagai unit waktu yang diharapkan dari penambangan 2016 blok yang valid.

Definisi 3: (*Pendapatan Absolut*) *Pendapatan absolut adalah jumlah rata-rata blok yang diperoleh di setiap waktu unit.*

1 Periode DAA. Kami memperlakukan periode DAA pertama sebagai awal penambangan egois untuk menganalisis pendapatan absolut sementara Alice dan Bob. Aturan normalisasi berikut dibuat untuk menyederhanakan analisis dengan menghilangkan keacakan interval penghasil blok. *Blok dihasilkan setiap unit waktu pada periode penyesuaian kesulitan pertama, yaitu,* Δ*t1* = 1.

Dengan asumsi di atas, kita dapat dengan mudah menghitung durasi menghasilkan blok valid 2016, meskipun sedikit mengorbankan ketelitiannya. Biarkan $_{Rvld}$ menjadi jumlah total blok yang valid dari semua penambang di putaran penambangan, dan biarkan $_{Rtot}$ menjadi jumlah total blok termasuk blok yang valid dan yatim piatu di babak yang sama. Ini berarti bahwa putaran penambangan menempati unit waktu $_{Rtot}$. Tunjukkan dengan *T1* satuan waktu yang diharapkan untuk mencapai periode DAA pertama. Seseorang dapat menghitung waktu putaran penambangan yang jumlah total blok yang valid mencapai 2016. Kemudian, *T1* setara dengan jumlah unit waktu putaran penambangan ini. Ada

$$R_{vld} = R1 + R2 + Rh;$$

$$R_{tot} = 1;$$

$$E[T_1] = \frac{2016}{R_{vld}} \cdot R_{tot}. \quad (31)$$

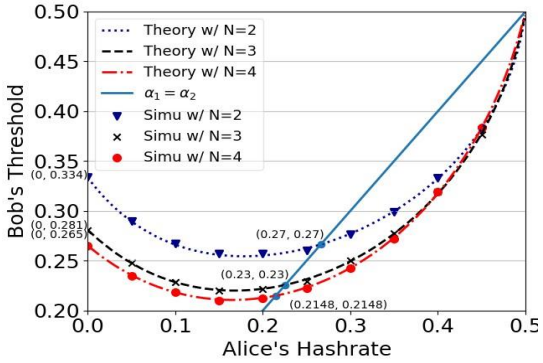
Karena blok yatim piatu, R_{tot} lebih besar dari R_{vld} sehingga waktu sebenarnya dari periode DAA pertama lebih lama dari unit waktu 2016.

Periode DAA berikutnya. Setelah periode DAA pertama, mekanisme konsensus blockchain menemukan bahwa interval waktu menghasilkan blok yang valid lebih dari satu unit waktu. Akibatnya, kesulitan target menurun agar sesuai dengan power Hash yang valid (terlihat atau terlihat) saat ini dalam sistem. Mengingat kekuatan Hash penambang yang tidak berubah, interval penghasil blok Δti menjadi lebih kecil untuk $i \geq 2$. Biarkan T_i menjadi unit waktu yang diharapkan dari periode DAA *ke-i* yang memiliki $Ti = 2016$ untuk $i \geq 2$. Perlu dicatat bahwa kita menganggap $R_{tot} \leq (4 \cdot Rvld)$. Ini adalah tujuan yang akan dicapai DAA.

Pendapatan absolut dari waktu ke waktu. Ingatlah bahwa pendapatan absolut menangkap hadiah yang diharapkan dari penambang di setiap unit waktu. Karena tujuan kami adalah untuk menyelidiki profitabilitas sementara penambangan egois, kami mendefinisikan $R_i(K)$ sebagai pendapatan absolut penambang *ke-i* selama periode *K* DAA. Oleh karena itu, kami memperoleh ungkapan berikut untuk $\forall i \in \{a,b,h\}$

$$\tilde{R}_i(K) = \frac{2016 \cdot K \cdot R_i}{R_{vld}} \cdot \frac{1}{\sum_{k=1}^{K} E[T_k]} = \frac{2016 \cdot KR_i}{R_{tot} + (K-1)R_{vld}}. \quad (32)$$

Sekarang kita sadar bahwa penambangan egois memiliki pendapatan absolut yang lebih kecil pada periode DAA pertama tidak peduli apakah kekuatan Hash penyerang berada di atas ambang batas stasioner yang menguntungkan atau tidak. Klaim ini juga berlaku dalam kebijakan penambangan egois yang berbeda. Ketika *K* meningkat, seorang penambang yang egois mudah-mudahan



Gambar 9. Ambang batas Bob di bawah pengaruh Alice's Hashrate.

dapat mengganti kerugiannya pada periode DAA pertama dengan pendapatan tambahannya di periode DAA mendatang. Dengan model pendapatan absolut kami, kami dapat menjelaskan berapa banyak waktu yang dibutuhkan untuk membuat penambangan egois menguntungkan pada akhirnya.

VI. EVALUASI

Pada bagian ini, kami mengembangkan simulator berbasis peristiwa untuk penambangan egois dasar dan simulator berbasis waktu untuk penambangan egois berbasis POMDP. Experiments komprehensif memvalidasi kebenaran model kami dan mengungkapkan sifat penting mengenai profitabilitas penambangan egois.

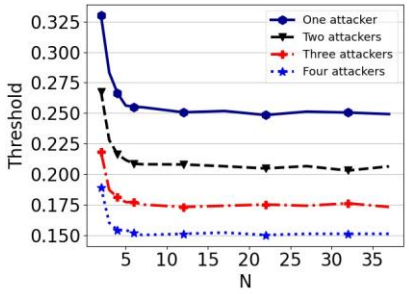
A. Pertambangan Egois Dasar

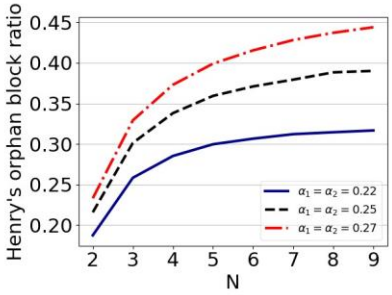
Pengamatan 1: Ketika ada beberapa penyerang dalam sistem seperti Bitcoin , ambang batas penyerang yang menguntungkan menurun dan keamanan sistem terdegradasi.

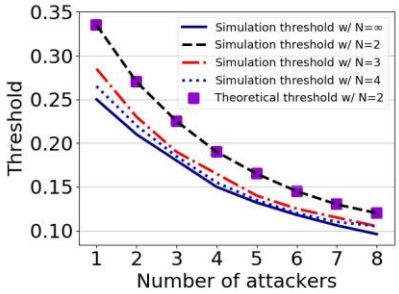
Kami mengilustrasikan ambang menguntungkan Bob dari penambangan egois di Gambar 9 saat kekuatan Hash Alice meningkat dari 0 menjadi hampir 0,5. Untuk menghindari melibatkan terlalu banyak variabel kontrol, parameter tiebreak diatur ke $\gamma_1 = \gamma_2 = \frac{1}{2}$ and $\theta_1 = \theta_2 = \frac{1}{3}$. Seseorang dapat mengamati dari tiga kurva dengan *N* yang berbeda bahwa ambang batas bob yang menguntungkan menurun di awal dan meningkat setelahnya. Terutama ketika $N = 2$ dan $\alpha_1 = 0.16$, ambang menguntungkan Bob adalah yang terendah. Dalam situasi ini, penambangan selfish Alice dapat menghasilkan lebih sedikit pendapatan dibandingkan dengan penambangannya yang jujur. Kami selanjutnya menggambar garis 45° untuk menunjukkan ambang batas yang menguntungkan untuk Alice dan Bob ketika kekuatan Hash mereka simetris. Ketika *N* adalah 2, 3 dan 4, ambang batas yang menguntungkan adalah 26,64%, 22,57% and 21,48%. Sebaliknya, dibutuhkan nilai masing-masing 33,33%, 28,08% dan 26,50%, jika ada penyerang tunggal. Kesimpulan yang jelas adalah bahwa keberadaan beberapa penyerang membuat penambangan egois lebih mudah menguntungkan.

Pengamatan 2: Ambang batas yang tidak dapat diterima menurun dengan peningkatan N, dan tetap stabil dengan penyerang kekuatan Hash simetris karena N ≥ 4; itu juga menurun ketika jumlah penyerang m meningkat.

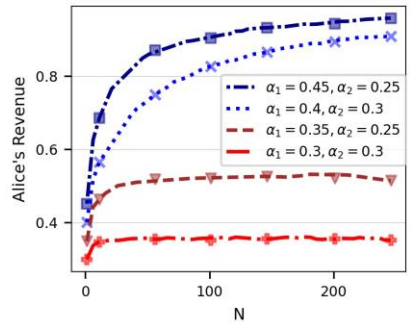
Kami mengevaluasi ambang batas BSM yang menguntungkan dengan *N* dan *m* yang berbeda. Tujuan kami ada dua: satu adalah menganalisis interaksi antara ambang batas ini dan variabel lingkungan, dan yang lainnya membenarkan penggunaan $N \leq 4$ dalam pemodelan matematika. Kekuatan Hash dari semua penyerang identik,

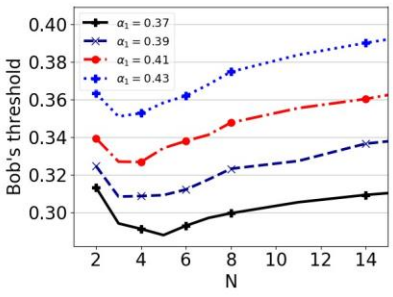


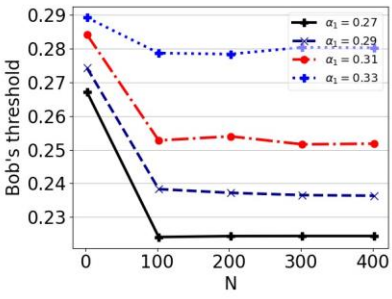




Gambar 10. Ambang batas yang menguntungkan vs jumlah maksimum- Gambar 11. Rasio blok yatim piatu Henry vs gambar maksimum 12. Ambang batas yang menguntungkan vs jumlah ber blok pribadi (*N*). jumlah blok pribadi (*N*). penyerang (*m*).







Gambar 13. Pendapatan Alice dengan $\alpha_1 >$ maks ($\alpha_{2,}\alpha_h$). Gambar 14. Ambang batas simulasi untuk Bob ketika $\alpha_1 >$ Gambar 15. Ambang batas simulasi untuk Bob saat $\alpha_h >$ maks($\alpha_{2,}\alpha_{jam}$). *maks*($\alpha_{1,}\alpha_2$).

dan rantai yang bersaing tidak dapat dibedakan pada aturan tiebreak.

Gambar 10 menunjukkan hubungan antara *N* dan ambang batas yang menguntungkan. Kasus-kasus dengan penyerang simetris 1, 2, 3 dan 4 dinyatakan dalam garis padat, putus-putus, putus-putus dan putus-putus. Orang dapat mengamati bahwa ambang batas yang menguntungkan sangat menurun untuk m yang berbeda karena *N* meningkat dari 2 menjadi 4. Simulasi yang digerakkan oleh acara menunjukkan ambang menguntungkan yang stabil ketika Alice dan Bob dapat menampung lebih dari 5 blok pribadi. Misalnya, threshold ini menyatu menjadi 25% untuk N yang cukup besar dengan penyerang tunggal, yang sejalan dengan [2]. Dengan dua penyerang simetris (*m* = 2), nilainya adalah 20,60% pada *N* = 30, sedikit berbeda dari pada *N* = 4. Secara umum, kekuatan Hash Alice atau Bob jauh lebih kecil daripada kekuatan Hash Henry. Kemungkinan bahwa rantai pribadi Alice atau Bob memimpin besar atas rantai publik di babak penambangan sangat kecil. Oleh karena itu, itu tidak membuat pengaruh yang jelas pada ambang batas yang menguntungkan ketika *N* sudah besar. Selain itu, menyembunyikan rantai pribadi yang panjang dan melepaskan semua blok secara bersamaan akan membuat serangan penambangan egois mudah dideteksi. Gambar 11 menunjukkan rasio blok yatim piatu Henry saat *N* meningkat dari 2 menjadi 9 dengan *m* = 2. Meskipun kekuatan Hash Alice dan Bob hanya (0,22,
0,22), mereka menyebabkan rasio blok yatim piatu yang sangat tinggi terhadap Henry, misalnya, 18,73% dengan *N* = 2, 28,53% dengan *N* = 4 dan 31,66% dengan *N* = 9. Rasio yatim piatu yang begitu tinggi dapat dengan mudah mengekspos identitas penyerang. Oleh karena itu, kerangka pemodelan kami hanya mempertimbangkan *N* ≤ 4 meskipun dapat diperluas ke *N* > 4.

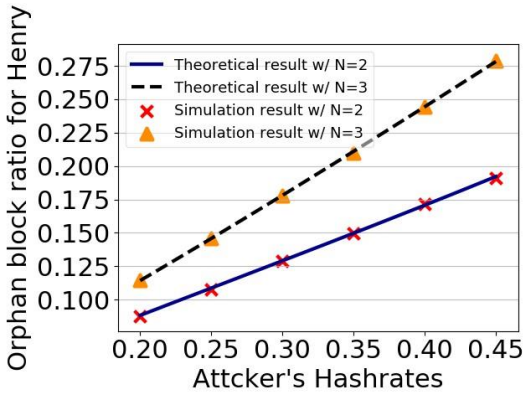
Kami menggunakan model matematika dan simulasi berbasis peristiwa untuk mengukur dampak *m* pada ambang batas yang menguntungkan pada Gambar 10 dan 12. Orang dapat mengamati bahwa increase dalam jumlah penyerang mengurangi ambang batas yang menguntungkan, sehingga membahayakan keamanan blockchain. Untuk *N* = 4, ambang batas yang menguntungkan dengan *m* ∈ {2,4,8} adalah {0. 2148,0. 155,0. 11} masing-masing. Ini menantang kognisi bahwa penambangan egois kurang disukai terjadi jika kekuatan Hash dari kolam penambangan di bawah 25%. Alasan utama bahwa lebih banyak penyerang menyebabkan ambang menguntungkan yang lebih kecil terletak pada bahwa kekuatan Hash dari penambang yang jujur menurun secara relatif. Sementara itu, model kami bertepatan dengan hasil simulasi pada *N* = 2 pada Gambar 12, sehingga memvalidasi kebenarannya.

Sejak munculnya pekerjaan mani, komunitas Bitcoin mencoba untuk membatasi kolam penambangan untuk memiliki kurang dari 25% dari kekuatan Hash. Namun, kami membuktikan bahwa 25% tidak *cukup*: Penambangan Bitcoin rapuh di hadapan beberapa penambang egois.

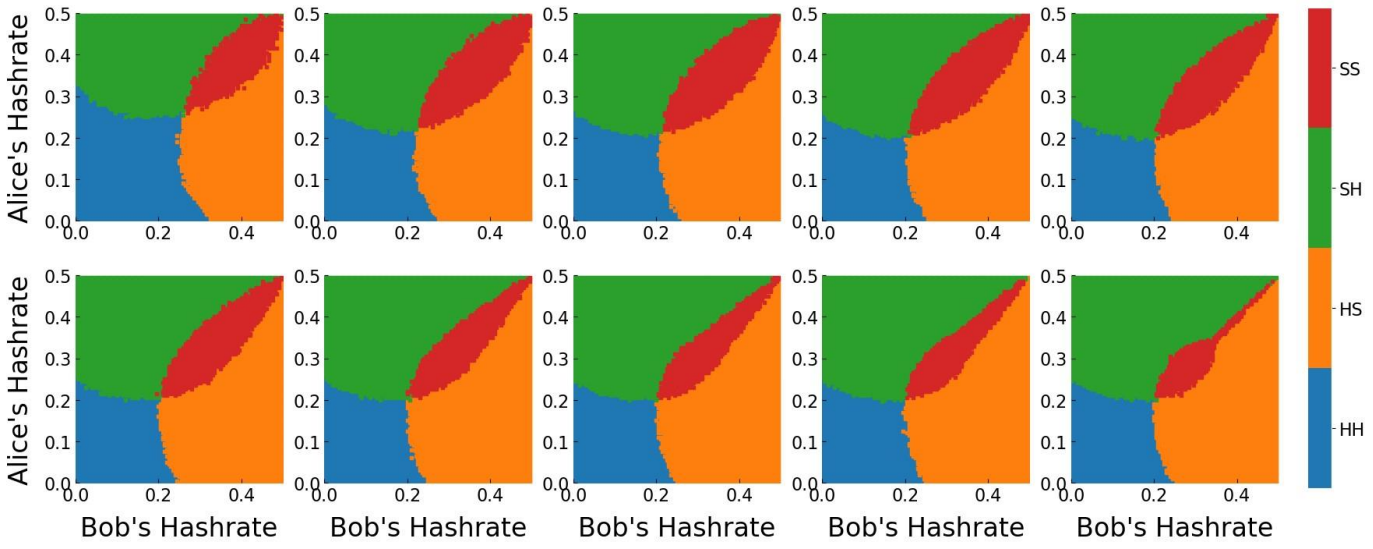
Pengamatan 3: Jika $\alpha_h <$ maks ($\alpha_{1,}\alpha_2$)," *pendapatan penyerang dengan kekuatan Hash lebih banyak di antara tiga*

penambang akan meningkat saat N meningkat. Jika α ₕ > maks (*α 1,α* ₂), *pendapatan penyerang dengan kekuatan Hash yang lebih banyak akan meningkat terlebih dahulu dan kemudian tetap stabil saat* N *meningkat. Manfaat umum Wilayah kekuatan Hash dari dua penyerang meningkat pada awalnya dan kemudian menurun dengan peningkatan N.*

Kami mengeksplorasi pendapatan para penyerang dengan kekuatan Hash lebih dari penambang jujur di bawah *N* yang berbeda ketika *α ₕ* < max (*α 1,α* ₂). Dalam Gambar 13, kami menunjukkan pendapatan Alice dari Alice pada empat situasi: kekuatan Hash Alice, Bob dan Henry adalah (0,45, 0,25, 0,3), (0,4, 0,3, 0,3), (0,35, 0,25, 0,4) dan (0,3, 0,3, 0,4) yang diberi label sebagai situasi 1, 2, 3 dan 4. Situasi 1 dan 2 menyatakan bahwa pendapatan penyerang dengan kekuatan Hash lebih banyak daripada yang lain meningkat saat *N* meningkat. Penyerang dapat memperoleh lebih dari 90% dari pendapatan, mencapai efek yang sama seperti serangan 51%, meskipun dia tidak memiliki 51% dari kekuatan Hash. Situasi 3 dan 4 menunjukkan bahwa pendapatan Alice menyatu saat *N* meningkat jika Henry memiliki kekuatan Hash terbesar. Pendapatan Alice menyatu menjadi 0,524 dalam situasi 3 dan 0,357 dalam situasi 4 dengan *N* = 200. Ini menyiratkan bahwa membatasi kekuatan Hash penyerang akan mencegah penyerang mendapatkan terlalu banyak pendapatan ketika *N* besar.



Kami kemudian menyelidiki ambang menguntungkan Bob when Alice memiliki kekuatan Hash yang berbeda dan *N* besar. Gambar 14 menunjukkan ambang batas Bob di bawah *N yang* berbeda ketika Alice memiliki kekuatan Hash terbesar, yaitu *α* ₁ > max{*α 2,α ₕ*}. Ambang menguntungkan Bob menurun terlebih dahulu dan kemudian meningkat saat *N* meningkat. Itu berarti bahkan jika Bob bisa mendapatkan penghasilan tambahan ketika *N* kecil, dia tidak bisa mendapatkan penghasilan tambahan ketika *N* cukup besar. Dalam keadaan ini, pendapatan Alice dapat memperoleh jauh lebih banyak daripada proporsi kekuatan Hash-nya, dan serangannya menjadi tidak berarti karena sistem ini tidak dapat menarik penambang lain bahkan penyerang penambangan egois lainnya. Gambar 15 menunjukkan ambang menguntungkan Bob akan berkurang terlebih dahulu dan kemudian bertemu ketika *N* meningkatkan when Henry memiliki lebih banyak kekuatan Hash daripada Alice dan Bob. Ini menunjukkan bahwa membatasi kekuatan Hash penyerang juga mendorong penambang lain termasuk penyerang pertambangan egois lainnya untuk terus menambang bahkan jika *N* besar. Menurut analisis ini, penyerang tidak akan menyembunyikan terlalu banyak blok bahkan dia memiliki lebih banyak kekuatan Hash daripada yang lain.

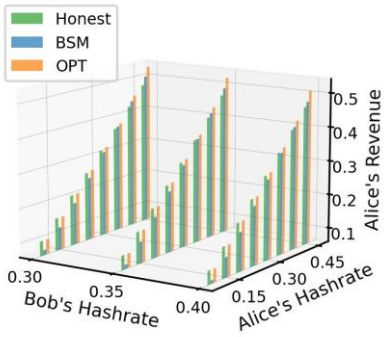


Gambar 16. Wilayah menguntungkan dari kedua penyerang penambangan selish dengan *N* = 2,3,4,5,7,8,15,25,35,∞.

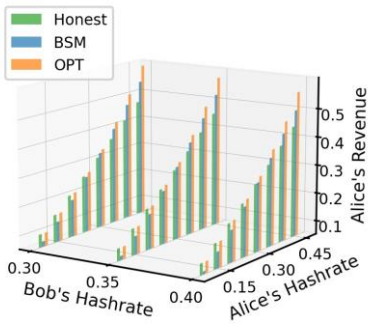
Gambar 17. Memperkirakan kekuatan Hash Bob dengan mengamati rasio blok yatim piatu.

Kami sekarang menganalisis interaksi antara ambang batas Alice dan Bob yang menguntungkan. Gambar 16 menunjukkan wilayah yang menguntungkan untuk setiap penyerang di bawah *N* yang berbeda. Bagian biru menunjukkan bahwa tidak ada penyerang yang dapat memperoleh pendapatan tambahan jika mereka melakukan serangan BSM, dan bagian merah menunjukkan bahwa kedua penyerang dapat memperoleh pendapatan tambahan melalui penambangan

egois. Bagian hijau (resp. orange) mewakili situasi bahwa hanya Alice (resp. Bob) yang dapat memperoleh pendapatan tambahan. Persimpangan empat wilayah sebenarnya adalah ambang batas yang menguntungkan dengan kekuatan Hash simetris Alice dan Bob yang menurun di atas *N* dan menyatu secara bertahap. Wilayah menguntungkan umum (merah) pertama-tama berkembang dan kemudian menyusut saat *N* meningkat. Alasannya adalah sebagai berikut. Sebuah wilayah merah besar pada dasarnya mengatakan bahwa baik Alice dan Bob menguntungkan dengan BSM bahkan jika kekuatan Hash mereka asimetris sampai batas tertentu. Ketika *N* sangat kecil, Alice dan Bob hanya dapat menyembunyikan beberapa blok sehingga kemampuan mereka untuk menyia-nyiakan kekuatan Hash Henry terkendali. Dengan meningkatnya *N*, kemampuan penambangan egois mereka menjadi lebih kuat, dan dengan demikian dapat memiliki lebih banyak peluang untuk mengeluarkan rantai publik bahkan jika masing-masing dari mereka memiliki kekuatan Hash yang lebih kecil daripada Henry. Sementara itu, mengingat pembatasan *N*, baik Alice dan Bob dapat menerima sejumlah pendapatan tambahan, menghasilkan wilayah menguntungkan umum yang lebih besar. Ketika *N* besar, penyerang yang lebih kuat cenderung mendominasi penambangan egois. Jika kekuatan Hash Alice lebih besar dari Bob, Bob akan merasa sulit untuk bersaing dengan Alice sehingga ambang menguntungkan Bob semakin tinggi. Jika kekuatan Hash Alice adalah yang terbesar, itu mirip dengan serangan 51%. Penambangan egois Bob hanya menguntungkan ketika kekuatan Hash Alice dan Bob cukup dekat, menyebabkan wilayah menguntungkan umum menyusut ke segmen garis.



Gambar 18. Pendapatan optimal untuk Alice ketika *N* = 2.



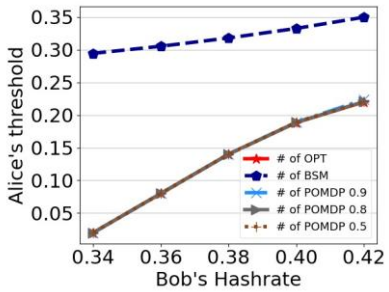
Gambar 19. Pendapatan optimal untuk Alice ketika *N* = 3.

B. MDP dan Pertambangan berbasis POMDP

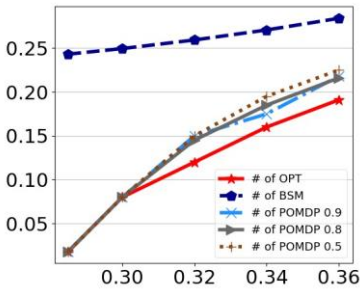
Peta jalan performing penambangan optimal adalah sebagai berikut. Kami mengeksplorasi kelayakannya dengan memperkirakan parameter yang tidak diketahui. Kemudian, kami menghitung kebijakan penambangan yang optimal dan pendapatan yang sesuai menggunakan MDP, dan atas dasar ini kami menghitung kebijakan penambangan optimal menggunakan POMDP di bawah status yang dapat diamati sebagian.

Ingatlah bahwa Alice adalah penambang optimal (OPT) dan Bob adalah penambang egois dasar (BSM). Pada tahap pertama, Alice perlu memutuskan apakah ada penambang egois yaitu Bob, dan jika demikian, apa kekuatan Hash Bob. Perhatikan that ada pemetaan satu-toone antara rasio blok yatim piatu Henry dan kekuatan Hash Bob. Gambar 17 menunjukkan rasio yatim piatu dari penambang jujur sebagai fungsi dari kekuatan Hash Bob dengan *N* = 2 dan *N* = 3. Hasil teoritis dan eksperimental cocok dengan baik, yang menunjukkan kelayakan menghitung kekuatan Hash Bob melalui rasio blok yatim piatu yang diamati.

Kami selanjutnya menghitung kebijakan optimal Alice dan pendapatan yang sesuai dari penambangan OPT. Biarkan parameter kesalahan $\epsilon = 0.00001$ dan nomor eksekusi ξ_0 = 500000. Gambar 18 menggambarkan pendapatan optimal yang diperoleh Alice ketika kekuatan Hash Alice $\alpha_1 \in \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45\}$, Kekuatan Hash Bob adalah $\alpha_2 \in \{0.30, 0.35, 0.40\}$ dan *N* = 2. Panjang maksimum rantai publik terpanjang ditetapkan sebagai (*N* +1). Orang dapat mengamati bahwa penambangan Alice yang optimal dan penambangan yang jujur menghasilkan pendapatan yang sama ketika kekuatan penambangannya relatif kecil, misalnya $\alpha_1 = 0.1$, $\alpha_2 = 0.40$ dan $\alpha_h = 0.50$. Dalam situasi ini, kebijakan penambangan yang optimal adalah penambangan yang jujur, sementara BSM berkinerja buruk pada penambangan yang jujur secara signifikan. Sebaliknya, ketika kekuatan Hash Alice adalah



Gambar 20. Ambang batas yang menguntungkan untuk Alice ketika *N* = 2.



Gambar 21. Ambang menguntungkan untuk Alice ketika $N = 3$.

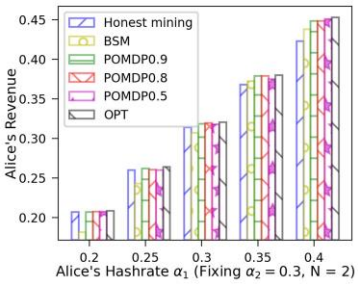
besar, misalnya, $\alpha_1 = 0.40$, $\alpha_2 = 0.40$ dan $\alpha_h = 0.20$, kebijakan pertambangan yang optimal menghasilkan pendapatan yang lebih tinggi daripada penambangan egois dasar, dan penambangan egois dasar lebih baik daripada penambangan yang jujur. Gambar 19 menunjukkan satu set percobaan similar kecuali N = 3. Ketika $\alpha_1 = 0.30$, $\alpha_2 = 0.30$ dan $\alpha_h = 0.4$, kebijakan penambangan Alice yang optimal jelas mengungguli BSM dan penambangan egois dasar.

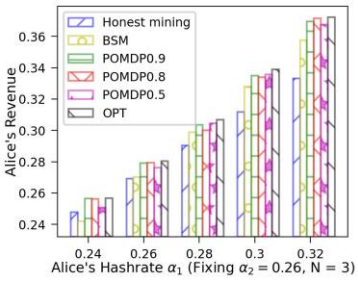
Kami menjelaskan contoh konkret dari kebijakan pertambangan yang optimal di beberapa negara perwakilan untuk kesederhanaan. Tabel I menunjukkan strategi optimal Alice dengan $\alpha_1 = 0.2$, $\alpha_2 = 0.4$ dan $\alpha_h = 0.4$. Alice memiliki kekuatan Hash yang lebih sedikit daripada Bob dan Henry. Dia memilih untuk mengadopsi rantai publik jika panjang rantainya lebih pendek dari Bob atau Henry. Jika panjang rantai (yaitu $l1 + h1$) dia menambang sama dengan rantai publik (yaitu $h3$), dan mampu "garpu", dia akan melepaskan *blok pribadi l1* untuk merebut kesempatan menang. Jika Alice dan Bob memegang blok pribadi, dia akan memilih untuk menyembunyikan blok pribadi dan terus menambang setelah bloknya sendiri. Dengan melakukan kebijakan optimal, Alice dapat memperoleh tambahan pendapatan 0,08%, meskipun Bob memiliki kekuatan Hash 40%.

TABEL I
KEBIJAKAN OPTIMAL UNTUK ($\alpha_1 = 0.2.\alpha_2 = 0.4$,N = 2).

negara	perbuatan
$(l2 + h2) > (l1 + h1)$	*Mengadopsi*
$h3 > (l1 + h1)$	*Mengadopsi*
$(l1 + h1) = (l2 + h2) =$ *h3, garpu* = r	$l1$
$l1 = l2 = 1.h3 = 0$	0
$l1 = 1.l2 = 0$	$l1$
$(l1 + h1) > (l2 + h2)$	$l1$

Pengamatan 4: Ketika Alice menggunakan kebijakan berbasis POMDP, dan Bob menggunakan penambangan egois dasar, Alice memiliki ambang menguntungkan yang jauh lebih rendah dibandingkan dengan penambangan egois dasar.

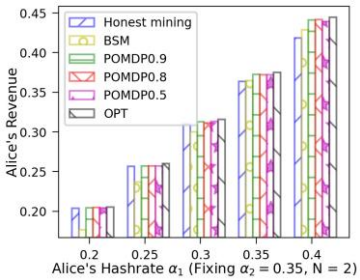




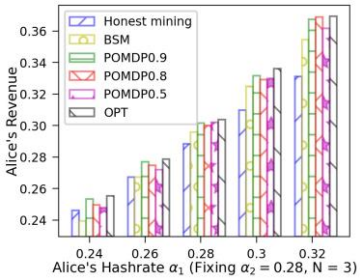
Pendapatannya tidak kurang dari penambangan yang jujur dan penambangan egois dasar , dan dekat dengan kebijakan berbasis MDP dengan informasi negara yang lengkap.

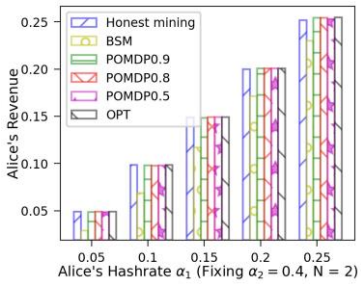
Kami selanjutnya mengevaluasi ambang batas dan pendapatan Alice yang menguntungkan ketika dia menggunakan kebijakan penambangan berbasis POMDP. Gambar 20 dan 21 membandingkan ambang menguntungkan strategi penambangan BSM, MDPbased (OPT) dan POMDP berbasis (POMDP) ketika $N = 2,3$ dan α_2 meningkat dari 0,285 menjadi 0,42. Temuan yang menarik adalah bahwa strategi OPT dan POMDP memiliki perontok menguntungkan yang jauh lebih kecil dibandingkan dengan BSM. Misalnya, ambang batas yang menguntungkan adalah 0,02 ketika $\alpha_2 = 0.34$ dan $N = 2$, dan 0,08 ketika $\alpha_2 = 0.3$ dan $N = 3$. Alasannya adalah sebagai berikut. Ketika Bob bermain BSM dan kekuatan Hash Bob jauh lebih besar dari Alice, Alice akan memilih to milikku dengan jujur. Jika ada garpu antara Bob dan Henry, Alice bersikeras menambang di rantai publik yang berisi bloknya sendiri. Kebijakan Alice setara dengan penurunan γ_2 dalam situasi penyerang tunggal. Oleh karena itu, Bob akan merasa sulit untuk mendapatkan lebih banyak pendapatan, dan Alice serta Henry mendapat manfaat dari kerugian Bob. Ketika kekuatan Hash Alice berada di bawah ambang batas yang menguntungkan, kebijakan POMDP tidak dapat menghasilkan pendapatan yang dapat dinikmati dengan kekuatan Hash-nya. Selain itu, dengan peningkatan kekuatan Hash Bob, ambang menguntungkan Alice menjadi lebih tinggi. Perbandingan silang antara Gambar 20 dan Gambar 21 menunjukkan bahwa N yang lebih besar membuat Alice sulit bersaing dengan Bob jika $\alpha_1 \leq \alpha_2$. Ketika kekuatan Hash Bob adalah 0,36, ambang menguntungkan Alice adalah 0,08 ketika $N = 2$ dan is sekitar 21,6% ketika $N = 3$.

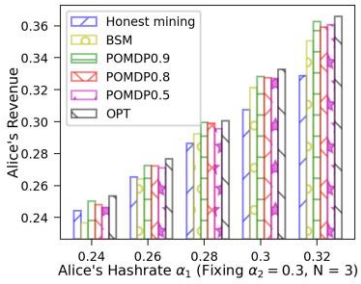
Kebijakan POMDP dapat meningkatkan pendapatan Alice dalam situasi yang berbeda. Gambar 22∼24 plot pendapatan



Alice menggunakan penambangan berbasis Honest mining, BSM, OPT dan POMDP di *N* = 2; Gambar 25∼27 menunjukkan pendapatan tersebut dengan *N* = 3. Di setiap set experiments, kami memperbaiki kekuatan Hash Bob (α_2) dan mengubah kekuatan Hash Alice (α_1) dari 0,20 menjadi 0,40. Adapun kebijakan POMDP, tiga kasus considered, di mana probabilitas menghasilkan blok adalah 0,9, 0,8 atau 0,5 pada setiap slot waktu. Perhatikan bahwa probabilitas ini tidak mempengaruhi eksekusi pemecah POMDP kami. Orang dapat dengan mudah mengamati bahwa kebijakan POMDP memiliki pendapatan yang sebanding dengan kebijakan penambangan yang jujur dan MDP ketika kekuatan Hash Alice relatif kecil. Sementara penambangan egois dasar tampaknya sangat "keras kepala", menyebabkan pendapatan Alice jauh lebih rendah daripada penambangan jujur dalam situasi ini. Kebijakan berbasis PODMP menghasilkan pendapatan yang sama atau lebih tinggi daripada penambangan yang jujur dan penambangan egois dasar, dan mendekati kinerja kebijakan MDP.







Algoritma online yang dirancang dapat secara efektif menghitung pendapatan optimal MOP. Kami membandingkan jumlah iterasi yang diperlukan untuk mengeksekusi algoritma pencarian biner di [7] dan Algoritma 1. Rasio reduksi dirangkum dalam Tabel II. Di bawah kombinasi kekuatan Hash yang berbeda dan slot aksi yang berbeda, efisiensi Algoritma 1 secara signifikan lebih tinggi. Ketika *N* = 2, $\alpha_1 = 0.3$, $\alpha_2 = 0.3$ dan $p = 0.5$, kita dapat menghemat 46% dari waktu komputasi. Butuh waktu lama untuk mensimulasikan setengah juta kali untuk mendapatkan pendapatan untuk setiap ρ. Peningkatan algoritma pencarian kami memainkan peran penting dalam memecahkan MPO dengan cepat.

C. Penambangan Egois pada Beberapa Periode Penyesuaian Kesulitan

Gambar 22. Pendapatan untuk Alice ketika α_2 0.= 3,N = 2.

Gambar 23. Pendapatan untuk Alice ketika α_2 0.= 35,N = 2.

Gambar 24. Pendapatan untuk Alice ketika α_2 0. 4,N= = 2.

Gambar 25. Pendapatan untuk Alice ketika α_2 0.= 26,N = 3.

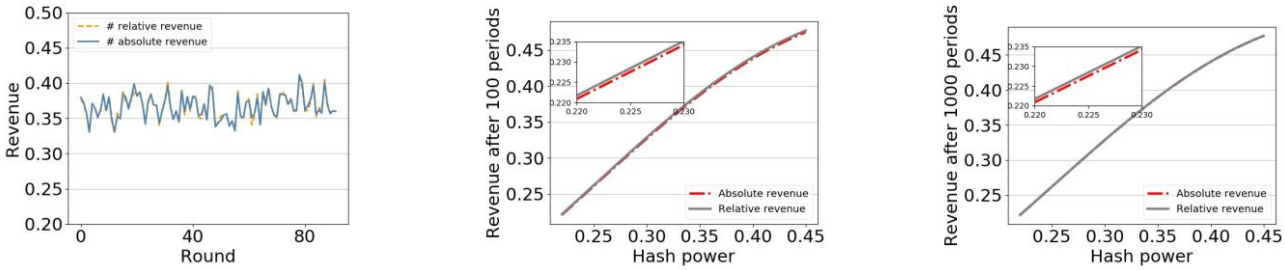
Gambar 26. Pendapatan untuk Alice ketika α_2 0.= 28,N = 3.

Gambar 27. Pendapatan untuk Alice ketika α_2 0. 3,N= = 3.

Pengamatan 5: Seorang penambang egois memperoleh pendapatan absolut yang lebih kecil daripada penambangan jujur selama periode penyesuaian kesulitan pertama terlepas dari hash power-nya. Namun, ia mungkin mendapatkan

keuntungan setelah sejumlah periode yang terkait dengan kekuatan Hash penambang yang egois.

Gambar 28 menunjukkan *pendapatan relatif* dan *pendapatan absolut* penyerang dengan Hashrate yang sama 33% dan *N* = 4 di setiap DAA

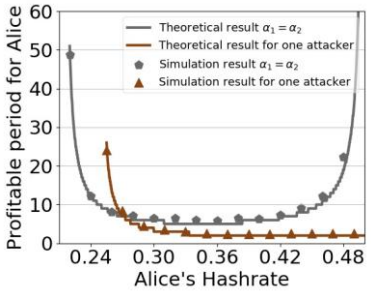


Gambar 28. Pendapatan relatif dan pendapatan absolut saat Gambar 29. Pendapatan relatif teoritis dan abso-Fig. 30. Pendapatan relatif teoritis dan abso α_1 $\alpha_2 = 0.$ 33,N = 4. pendapatan kecapi saat $\alpha_1 = \alpha_2$ setelah 100 periode α α.

TABEL II

RASIO PENGURANGAN WAKTU ALG. 1 DIBANDINGKAN DENGAN ALGORITMA TRADISIONAL.

Kekuatan hash	*p*	EFF IMP
$_{a1} = 0.$ *3,a2* = 0. 3 *N* = 2	0.9	53.3%
	0.8	46.6%
	0.5	66.6%
$_{a1} = 0.$ 35,a2 = 0. 35 *N* = 2	0.9	80%
	0.8	60%
	0.5	46.7%
$_{a1} = 0.$ 28,a2 = 0. 28 *N* = 3	0.9	60%
	0.8	46.7%
	0.5	46.7%
$_{a1} = 0.$ *3,a2* = 0. 3 *N* = 3	0.9	73.3%
	0.8	46.7%
	0.5	40%



Gambar 31. Waktu yang menguntungkan dan kekuatan Hash.

masa. *Pendapatan relatif* dan *pendapatan absolut* sama dalam rentang kesalahan yang diizinkan. Oleh karena itu, *pendapatan relatif* dapat memainkan aturan yang sama dengan *pendapatan absolut* dalam mewakili manfaat. Gambar 29 dan Gambar 30 menunjukkan *pendapatan relatif* teoritis dan *pendapatan absolut* setelah 100 dan 1000 period DAA ketika $\alpha_1 = \alpha_2$. Dapat diamati bahwa *pendapatan absolut* selalu kurang dari *pendapatan relatif*, tetapi perbedaannya kecil. Saat $\alpha_1 =$ 0. 22, *pendapatan relatif* adalah 0,2217 dan *pendapatan absolut* adalah 0,2209. Perbedaan ini menurun menjadi 0,0001 after 1000 periode. Itu berarti perbedaan antara *pendapatan relatif* dan *pendapatan absolut* menurun dengan peningkatan waktu serangan.

Seperti yang ditunjukkan Eq. (32), ketika Alice memiliki lebih banyak kekuatan Hash, dia bisa mendapatkan pendapatan ilegal lebih awal. Namun, jika kedua attacker memiliki Hashrate besar, mereka akan mendapat manfaat terlambat. Gambar 31 menunjukkan hasil simulasi dan hasil teoritis waktu yang menguntungkan di bawah kekuatan Hash yang berbeda dengan penyerang simetris. Sumbu horizontal mewakili kekuatan Hash serangan dan hiasan mewakili waktu yang menguntungkan penyerang, juga kurva adalah hasil teoritis dan titik-titik adalah hasil simulasi. Penambangan egois menguntungkan setelah 51 putaran penyesuaian kesulitan (yaitu 714 hari dalam Bitcoin) jika Hashrates dari miner egois keduanya 22% (sedikit lebih tinggi dari ambang

menguntungkan). Penundaan ini menurun menjadi 5 putaran (yaitu 70 hari dalam Bitcoin) karena Hashrates mereka bertambah menjadi 33%, yang masih sangat panjang. Ketika para penyerang mendapatkan lebih banyak daya komputasi, pesaing utama Alice menjadi Bob daripada Henry. Waktu manfaat mulai meningkat untuk Alice dan Bob. Ketika hanya ada satu penyerang dan dia memiliki Hashrates 25,5%, penyerang akan mendapatkan pendapatan tambahan setelah 26 putaran (sekitar 364 hari).

Ini menunjukkan bahwa ketika Hashrate penyerang relatif kecil, dibutuhkan waktu yang agak lama untuk mendapatkan keuntungan. Ketika kedua penyerang semuanya memiliki Hashrate besar, juga butuh waktu lama untuk mendapatkan pendapatan tambahan. Itu berarti dalam sistem nyata, agak sulit untuk melakukan serangan. Jika global Hashrate meningkat, kita juga dapat menggunakan rumus ini untuk menghitung kapan harus menghentikan serangan sebelum kita bisa mendapatkan keuntungan paling banyak.

VII. PEKERJAAN TERKAIT

Serangan penambangan egois adalah salah satu tantangan inti dari konsensus blockchain yang telah dipelajari secara ekstensif dalam beberapa tahun terakhir. Kami dengan ini menjelaskan kemajuan terbaru dalam kebijakan pertambangan egois dan kinerja analitis atau eksperimental mereka.

Satu penyerang. Sejak munculnya [2], ada banyak penelitian tentang berbagai bentuk serangan penambangan egois. Nayak et al. [9] mengusulkan penambangan keras kepala berdasarkan penambangan egois dasar, ini menunjukkan bahwa penambangan egois tidak selalu yang terbaik untuk parameter yang berbeda. Serangan keras kepala meningkatkan sekitar 13,94% pendapatan dibandingkan dengan serangan penambangan egois dasar. Strategi penambangan egois yang lebih cerdas telah diusulkan dalam [7] berdasarkan Proses Keputusan Markov dan menurunkan ambang batas yang menguntungkan menjadi 23,21%. Tao et al. [20] menggambarkan serangan penambangan semi-egois atas dasar penambangan egois berdasarkan proses keputusan Markov yang tersembunyi, yang tidak hanya memastikan manfaat dari serangan itu, tetapi juga mengurangi tingkat forking. Negy et al. [23] memperkenalkan *penambangan egois intermittent* dan menunjukkan bahwa penambang egois intermitt di atas 37% kekuatan hash menghasilkan lebih banyak koin per unit waktu bahkan ketika $\gamma = 0$. Negy dan Davidson et al. [23] [24] mensimulasikan profitabilitas penambangan egois di bawah beberapa kesulitan penyesuaian algorithms yang digunakan dalam cryptocurrency populer. Serangan penambangan egois mengambil properti yang berbeda dalam sistem Ethereum karena blok paman. Grunspan dan Ritz memperkenalkan serangan penambangan egois di Ethereum dan menemukan bahwa Ethereum lebih rentan terhadap penambangan selfish daripada Bitcoin [25] [26].

Pada saat yang sama, serangan penambangan egois juga dapat dikombinasikan dengan serangan lain untuk mencapai manfaat yang lebih besar. Gervais dkk. merancang strategi optimal untuk pengeluaran ganda dan penambangan egois sambil mempertimbangkan kendala dunia real seperti propagasi jaringan, ukuran blok yang berbeda, interval generasi blok, mekanisme propagasi informasi, dan dampak serangan gerhana [22]. Kwon et al. [14] mengusulkan serangan FAW yang menggabungkan serangan penambangan egois dan menahan serangan. Hadiah untuk penyerang FAW selalu sama dengan atau lebih besar dari itu untuk penyerang BWH. Gao dkk. memperpanjang pekerjaan Kwon dkk. mengusulkan power adjusting withholding attack (PAW) dan bribery selfish mining attack (BSM) [15]. Mereka menunjukkan PAW dapat menghindari "dilema penambang" dalam serangan BWH dan BSW meningkatkan pendapatan 10% dibandingkan dengan SM tradisional[33]. Namun, BSW akan memperkenalkan "dilema penambang venal". Untuk menghindari "dilema penambang venal", Yang dkk. mengusulkan attack IPBSM yang mengasumsikan semua penyerang mengambil penambangan egois penyuapan optimal [16].

Beberapa penyerang. Partisipasi beberapa penyerang dalam sistem akan sangat mengubah manfaat dari serangan penambangan yang egois. Ruan dkk. mensimulasikan situasi di mana ada dua penyerang dalam sistem dan memperoleh ambang batas yang sesuai akan berkurang [3]. Francisco dkk. mengusulkan penambangan semiselfish ketika ada dua penyerang dan mencontoh serangan ini [17]. Mereka memperoleh ekuilibrium Nash di bawah kekuatan Hash yang berbeda dan ambang batas untuk setiap kebijakan berdasarkan teori permainan. Juga, mereka memodelkan situasi ketika jumlah penyerang lebih dari dua. Namun, mereka tidak mendapatkan hasil bentuk tertutup. Azimy et al. merancang simulator jaringan Bitcoin dan menggunakannya untuk mensimulasikan konfigurasi penambang yang berbeda untuk dapat mengatasi masalah ini. Temuan mereka menunjukkan bahwa di hampir semua konfigurasi, dengan kehadiran penambang egois yang lebih kuat, penambangan egois benar-benar mengurangi pendapatan para penambang egois yang menangis dan juga membantu penambang egois yang lebih kuat. Sebastian dkk. mengusulkan hasil simulasi bahwa ambang batas yang menguntungkan menurun sebanding dengan jumlah penambang egois [35]. Selain itu, ada ekuilibrium Nash di mana penambang egois menambang dengan jujur dan bersamaan mendapatkan hadiah penambangan yang tidak adil. Zhang dkk. mensimulasikan situasi ketika ada beberapa penyerang penambangan egois dalam sistem. Ini menunjukkan ada skenario di mana cukup untuk memiliki 12% power pertambangan untuk mendapatkan keuntungan dari penambangan egois tetapi juga bahwa memiliki lebih dari tujuh penambang egois yang mendapat manfaat secara bersamaan sangat tidak mungkin [18]. Charlie dkk. mengusulkan SquirRL yang merupakan kerangka kerja untuk menggunakan pembelajaran penguatan mendalam

untuk menganalisis serangan terhadap mekanisme insentif blockchain. Pendapatan SquirRL lebih besar daripada penyerang keputusan Markov ketika ada beberapa penyerang pertambangan egois[19]. Xia et al. mengeksplorasi dampak dari beberapa penambang dan penundaan propagasi pada penambangan egois [29].

VIII.C ONLUSI

Dalam makalah ini, pertama-tama kita mempelajari bagaimana keberadaan beberapa penambang yang berperilaku buruk mempengaruhi profitabilitas penambangan egois dasar. Dengan menetapkan model rantai Markov untuk menggambarkan aksi penyerang dan penambang yang jujur, kita dapat obtain ambang batas minimum yang menguntungkan adalah simetris 21,48%, yang menurun karena jumlah penyerang simetris meningkat. Jika ada dua penyerang penambangan egois asimetris dalam sistem, ambang menguntungkan dari satu penyerang menurun terlebih dahulu dan kemudian muncul dengan peningkatan kekuatan Hash penyerang lainnya. Kami memvalidasi ini di kedua model dan eksperimen. Kami selanjutnya menyelidiki pendapatan para penyerang ketika yang satu mengeksekusi penambangan egois dasar dan yang lainnya menerapkan penambangan strategis. Strategi penambangan new dirancang untuk para penambang dengan informasi yang tidak lengkap berdasarkan *POMDP*. Kita dapat memperoleh pendapatan dengan strategi baru tidak kurang dari penambangan yang jujur dan penambangan egois dasar. Mempertimbangkan penyesuaian kesulitan, kami memodelkan proses sementara dan memperoleh solusi bentuk tertutup dari waktu yang menguntungkan. Dapat ditemukan bahwa waktu yang menguntungkan besar ketika kekuatan Hash penyerang rendah. Selain itu, ada korelasi negatif antara waktu yang menguntungkan dan kekuatan penambangan penyerang.

REFERENCES

[1] S. Nakamoto. "Bitcoin: Sistem uang elektronik peer-to-peer", 2008.

[2] I. Eyal dan E. G. Sirer. "Mayoritas tidak cukup: penambangan Bitcoin rentan." *Dalam Kriptografi Keuangan dan Keamanan Data.* Springer, 2014, hlm. 436-454.

[3] Q.H. Liu, N. Ruan, dkk. "Tentang Strategi dan Perilaku Penambangan Bitcoin dengan penyerang-N". *Proc. konferensi Asia tentang keamanan komputer dan komunikasi,* hlm. 357-368, 2018.

[4] https://decrypt.co/35373/how-long-does-it-take-to-mine-a-bitcoin, [online].

[5] A. Papoulis, S. U. Pillai. Probabilitas, variabel acak, dan proses stokastik [M]. Pendidikan Tata McGraw-Hill, 2002.

[6] R. Pass, E. Shi, "Fruitchains: Blockchain yang adil". *Konferensi Asia tentang Simposium Komputer tentang Prinsip-Prinsip Computing Terdistribusi,* hlm. 315-324, 2017.

[7] A. Sapirshtein, Y. Sompolinsky, A. Zohar. "Strategi penambangan egois yang optimal dalam bitcoin". *Konferensi Internasional tentang Kriptografi Keuangan dan Keamanan Data,* hlm. 515-532, 2016.

[8] S. Jiang dan J. Wu, "Penambangan Bitcoin dengan Biaya Transaksi: Permainan pada Ukuran Blok," *dalam Konferensi Internasional IEEE 2019 tentang Blockchain (Blockchain),* Atlanta, GA, AS, Juli 2019, hlm. 107–115.

[9] K. Nayak, S. Kumar, A. Miller, dan E. Shi, "Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack," *dalam 2016 IEEE European Symposium on Security and Privacy (EuroS&P),* Saarbrucken, Mar. 2016, hlm. 305–320.

[10] Puterman, Martin L. Markov proses keputusan: pemrograman dinamis stokastik diskrit. Putra John Wiley, 2014.

[11] Ross, Stephane, dan Brahim Chaib-Draa. "AEMS: Algoritma pencarian online kapan saja untuk perkiraan penyempurnaan kebijakan dalam POMDP besar." *IJCAI.* 2007, hlm. 2592-2598.

[12] Mereka adalah Karkus, Peter, David Hsu, dan Wee Sun Lee. "QMDP-Net: pembelajaran mendalam untuk perencanaan di bawah observabilitas parsial."

Proceedings dari Konferensi Internasional ke-31 tentang Sistem Pemrosesan Informasi Saraf. 2017.

[13] P. Ashok, K. Chatterjee, P. Daca, J. Kˇret'ınsky, dan T. Meggendorfer,' "Iterasi Nilai untuk Hadiah Rata-Rata Jangka Panjang dalam Proses Keputusan Markov," *dalam Verifikasi Berbantuan Komputer,* vol. 10426, Springer International Publishing, 2017, hlm. 201–221.

[14] Y. Kwon, D. Kim, Y. Son, E. Vasserman, dan Y. Kim, "Jadilah Egois dan Hindari Dilema: Garpu Setelah Menahan (FAW) Serangan terhadap Bitcoin," *dalam Prosiding Konferensi SIGSAC ACM 2017 tentang Keamanan Komputer dan Komunikasi,* Dallas Texas USA, Oktober 2017, hlm. 195–209.

[15] S. Gao, Z. Li, Z. Peng, dan B. Xiao, "Power Adjusting and Bribery Racing: Novel Mining Attacks in the Bitcoin System," *dalam Prosiding Konferensi SIGSAC ACM 2019 tentang Keamanan Komputer dan Komunikasi,* London Inggris, November 2019, hlm. 833–850.

[16] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, dan S. Li, "IPBSM: An . penyuapan optimal penambangan egois di hadapan penyerang cerdas dan murni," *Int J Intell Syst,* vol. 35, no. 11, hlm. 1735–1748, November 2020.

[17] F. J. Marmolejo-Coss'ıo, E. Brigham, B. Sela, dan J. Katz, "Bersaing (Semi-)Penambang Egois di Bitcoin," *dalam Prosiding Konferensi ACM ke-1 tentang Kemajuan Teknologi Keuangan,* Zurich Swiss, Oktober 2019, hlm. 89–109.

[18] S. Zhang, K. Zhang, dan B. Kemme, "Menganalisis Manfaat Penambangan Egois dengan Banyak Pemain," *dalam 2020 IEEE International Conference on Blockchain (Blockchain),* Rhodes Island, Yunani, November 2020, hlm. 36–44.

[19] Hou, Charlie, et al. "SquirRL: Mengotomatisasi analisis serangan pada mekanisme insentif blockchain dengan pembelajaran penguatan yang mendalam." *arXiv* 2019.

[20] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, dan X. Yu, "Penambangan semi-egois berdasarkan proses keputusan Markov yang tersembunyi," *Int J Intell Syst,* vol. 36, no. 7, hlm. 3596–3612, Jul 2021.

[21] Mereka adalah Meyn, Sean P., dan Richard L. Tweedie. *Rantai Markov dan stabilitas stokastik.* Springer Sains & Media Bisnis, 2012.

[22] A. Gervais, et al., "Tentang Keamanan dan Kinerja Proof of Work Blockchains," *dalam Prosiding Konferensi SIGSAC ACM 2016 tentang Keamanan Komputer dan Komunikasi*, Wina Austria, Oktober 2016, hlm. 3–16.

[23] K. A. Negy, P. R. Rizun, dan E. G. Sirer, "Selfish Mining Re-Examined," *dalam Kriptografi Keuangan dan Keamanan Data*, vol. 12059, J. Bonneau dan N. Heninger, Eds. Cham: Springer International Publishing, 2020, hlm. 61–78.

[24] Davidson, Michael, dan Tyler Diamond. "Tentang Profitabilitas Penambangan Egois Terhadap Beberapa Algoritma Penyesuaian Kesulitan." *IACR Cryptol.* Lengkungan *ePrint.* 2020 (2020): 94.

[25] C. Grunspan dan R. Perez-Marco, "Selfish Mining in Ethereum," *dalam Mathematical Research for Blockchain Economy,* P. Pardalos, aku. Kotsireas, Y. Guo, dan W. Knottenbelt, Eds. Cham: Springer International Publishing, 2020, hlm. 65–90.

[26] F. Ritz dan A. Zugenmaier, "Dampak Hadiah Paman pada Penambangan Egois di Ethereum," dalam Simposium *Eropa IEEE 2018* tentang *Lokakarya Keamanan dan Privasi (EuroS &PW),* London, April 2018, hlm. 50–57.

[27] https://tradeblock.com/bitcoin/historical.

[28] S. Ross, J. Pineau, S. Paquet, dan B. Chaib-draa, "Algoritma Perencanaan Online untuk POMDP", *jair,* vol. 32, hlm. 663–704, Juli 2008.

[29] Q. Xia et al., "Analisis Dampak Beberapa Penambang dan Penundaan Propagasi pada Penambangan Egois", *pada tahun 2021 Konferensi Komputer, Perangkat Lunak, dan Aplikasi Tahunan IEEE ke-45 (COMPSAC)*, Madrid, Spanyol, Juli 2021, hlm. 694–703.

[30] H. Azimy dan A. Ghorbani, "Competitive Selfish Mining", *dalam Konferensi Internasional ke-17 tentang Privasi, Keamanan dan Kepercayaan (PST) ke-17 tahun 2019,* Fredericton, NB, Kanada, 1-8 Agustus 2019.

[31] Q. Bai, X. Zhou, X. Wang, Y. Xu, X. Wang, dan Q. Kong, "A Deep Dive Into Blockchain Selfish Mining", *dalam ICC 2019 - 2019 IEEE International Conference on Communications (ICC),* Shanghai, China, Mei 2019, hlm. 1–6.

[32] C. Grunspan dan R. Perez-Marco, "Tentang profitabilitas penambangan egois", *arXiv,* 2019.

[33] I. Eyal, "Dilema Penambang", *arXiv*, 2014.
[34] X. Dong, F. Wu, A. Faree, D. Guo, Y. Shen, dan J. Ma, "Selfholding: Model serangan gabungan menggunakan penambangan egois dengan serangan pemotongan blok," *Computers* & *Security,* vol. 87, p. 101584, November 2019.
[35] T. Leelavimolsilp, L. Tran-Thanh, dan S. Stein, "Pada Penyelidikan Awal Strategi Penambangan Egois dengan Beberapa Penambang Egois," *arxiv*, 2018.
[36] Littman et al., "Kebijakan pembelajaran untuk lingkungan yang dapat diamati sebagian: meningkatkan". *Dalam Prosiding Konferensi Internasional ke-12 tentang Pembelajaran Mesin (ICML-95),* hlm. 362–370.

APPENDIX

TABEL III

DESKRIPSI TRANSISI DAN PENGHARGAAN MATRIKS P DAN R DALAM MASALAH KEPUTUSAN *M*.

	Mengadopsi	*tempat*! = 2	$_{a1}a2$ *ah*	(1,ir,1,l2,h3,h2,h3,μ $_{3,\mu\ 2,\mu}$ 3) (*1,ir,0,l2* + *1,h3,h2,h3,μ* $_{3,\mu\ 2,\mu\ 3}$) (1,r,0,l2,h3,h2,h3 $_{+\ 1.\mu\ 3.\mu\ 2.\mu}$ 3 + 1)	(*0,0,0*)
		tempat = 2	$_{a1}a2$ *ah*	(1,ir,1,l2,h3- $_{h2,0,h3}$− *h2,h3*− *h2,0,h3*− *h2*) (*1,ir,0,l2* + *1,h3*− h2,0,h3− $_{h2,h3}$− *h2,0,h3*− *h2*) (1,r,0,l2,h3- $_{h2,0,h3}$− *h2* + *1,h3*− $_{h2,0,h3}$− *h2* + 1)	(*0.h2*− *μ 2.μ* $_{2}$)
	$_{h1}$ + *aksi* < $_{h3}$		$_{a1}a2$ *ah*	(*loc,ir, l1*− *action* + 1,l2,h1 + share,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*loc,ir,l1*− *action,l2* + *1,h1* + share,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*loc,r,l1*− *action,l2,h1* + *share,h2,h3* + *1,μ 1,μ 2,μ* $_{3}$ + 1)	(*0,0,0*)
	$_{h1}$ + *aksi* = $_{h3}$	*r,aksi* > 0 $_{f13,action}$ = 0	$_{a1}a2$ *αhγ1 αh*(1 − *c1*)	(*loc,f13,l1*− *aksi* + 1,l2,h3,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*loc,f13,l1*− *action,l2* + *1,h3,h2,h3,μ 1,μ 2,μ* $_{3}$) (*1,r,l1*− share,l2,h3,h2,h3 $_{+\ 1,\mu\ 1,\mu\ 2,\mu\ 1}$ + 1) (*1,r,l1*− share,l2,h3,h2,h3 $_{+\ 1,\mu\ 1,\mu\ 2,\mu\ 3}$ + 1)	(*0,0,0*)
$_{l2}$ + *h2*− *h3* > 2	$_{h3}$ < *h1* + *aksi* < *l2* + *h2*− 1		$_{a1}a2$ *ahh*	(*1,ir,l1*− *share* + 1,l2,h1 + *share,h2,h1* + *share,μ 1,μ 2,μ* $_{1}$) (*1,ir,l1*− *share,l2* + *1,h1* + *share,h2,h1* + *share,μ 1,μ 2,μ* 1) (*1,r,l1*− *share,l2,h1* + *share,h2,h1* + *share* + *1.μ 1.μ 2.μ* $_{1}$ + 1)	(*0,0,0*)
	$_{h1}$ + *aksi* = *l2* + *h2*− 1		$_{a1}a2$ *ahh*	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$ $(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$ (*2,r,l1*− *share,0,h1* + *share,h2* + *l2* + *1,h2* + $_{l2}$ + *1,μ 1,μ* $_{2}$ + *1,μ* $_{2}$ + 1)	(*0,0,0*)
	$_{h1}$ + *aksi* = *l2* + *h2*		$_{a1}a2$ *ahb1*	(*loc,f12,l1*− *aksi* + 1,0,h1 + *share,h2* + $_{l2,h2}$ + $_{l2,\mu}$ *1,μ 2,μ* 3) (*1,r,l1*− *share,0,h1* + *share,h2* + *l2* + *1,h2* + $_{l2}$ + *1,μ 1,μ 2,μ* 2) (*2,r,l1*− *share,0,h1* + *share,h2* + *l2* + *1,h2* + $_{l2}$ + *1,μ 1,μ* $_{2}$ + *1,μ* $_{2}$ + 1)	(*0,0,0*)
			ahb2	(*2,r,l1*− aksi,0,0,1,1,0,1,1)	(*h1* + *aksi* - *μ 1,0,μ* $_{1}$)
	$_{h1}$ + *aksi* > *l2* + *h2*		$_{a1}a2$ *ahh*	(*3,ir,l1*− *saham* + 1,0,0,0,0,0,0,0,0) (*3,ir,l1*− aksi,1,0,0,0,0,0,0) ((*3,r, l1*− aksi,0,0,1,1,0,1,1))	(h1 + *aksi* - *μ 1.0.μ* $_{1}$)
	Mengadopsi	*tempat*! = 2	$_{a1}a2$ *ah*	(1,ir,1,l2,h3,h2,h3,μ $_{3,\mu\ 2,\mu}$ 3) (*1,ir,0,l2* + *1,h3,h2,h3,μ* $_{3,\mu\ 2,\mu\ 3}$) (2,r,0,0,h3,h2 $_{+\ l2,h2}$ + $_{l2,\mu}$ *3,μ 2,μ* $_{2}$)	(*0,0,0*)
		tempat = 2	$_{a1}a2$ *ah*	(1,ir,1,l2,h3- $_{h2,0,h3}$− *h2,h3*− *h2,0,h3*− *h2*) (*1,ir,0,l2* + *1,h3*− h2,0,h3− *h2,h3*− h2,0,h3− $_{h2}$) (*2,r,0,0,h3* - h2,l2,l2,h3- *h2,0,0*)	(*0.h2*− *μ 2.μ* $_{2}$)
	$_{h1}$ + *aksi* < $_{h3}$		$_{a1}a2$ *ah*	(*loc,ir, l1*− *action* + 1,l2,h1 + share,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*loc,ir,l1*− *action,l2* + *1,h1* + share,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*2,r,l1*− *share,0,h1* + *share,h2* + $_{l2,h2}$ + $_{l2,\mu}$ *1,μ 2,μ* 2)	(*0,0,0*)
$_{l2}$ + *h2*− *h3* = 2	$_{h1}$ + *aksi* = $_{h3}$	*r,aksi* > 0 $_{f13,action}$ = 0	$_{a1}a2$ *ah*	(*loc,f13,l1*− *aksi* + 1,l2,h3,h2,h3,μ $_{1,\mu\ 2,\mu}$ 3) (*loc,f13,l1*− *action,l2* + *1,h3,h2,h3,μ 1,μ 2,μ* $_{3}$)	(*0,0,0*)

				(2,r,l1− share,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 2)	
	h1 + aksi = l2 + h2− 1		a1 a2 ahh	$(2, r, l_1 + 1 - action, 0, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$ $(2, r, l_1 - action, 1, h_1 + action, h_2 + l_2, h_2 + l_2, \mu_1, \mu_2, \mu_2)$ (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2 + 1,μ 2 + 1)	(0,0,0)
	h1 + aksi = l2 + h2		a1 a2 ahb2	(loc,f12,l1− aksi + 1,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (1,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2,μ 2) (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2 + 1,μ 2 + 1)	(0,0,0)
			ahb1	(2,r,l1− aksi,0,0,1,1,0,1,1)	(h1 + aksi - μ 1,0,μ 1)
	h1 + aksi > l2 + h2		a1 a2 ahh	(3,ir,l1− saham + 1,0,0,0,0,0,0,0,0) (3,ir,l1− aksi,1,0,0,0,0,0,0) ((3,r, l1− aksi,0,0,1,1,0,1,1))	(h1 + aksi - μ 1.0.μ 1)
l2 + h2− h3 = 1	Mengadopsi	tempat = 2,tempat = 3	a1 a2 ah	(1,ir,1,l2,h3- h2,0,h3− h2,h3− h2,0,h3− h2) (1,ir,0,l2 + 1,h3− h2,0,h3− h2,h3− h2,0,h3− h2) (loc,f23,0,0 ,h3− h2,h3− h2 + 1,h3− h2 + 1,0,0,1)	(0.h2− μ 2.μ 2)
	h1 + aksi < h3		a1 a2 ah	(loc,ir, l1− action + 1,l2,h1 + share,h2,h3,μ 1,μ 2,μ 3) (loc,ir,l1− action,l2 + 1,h1 + share,h2,h3,μ 1,μ 2,μ 3) (loc,f23,l1− action,0,h1 + share,h3 + 1,h3 + 1,μ 1,μ 2,μ 3 + 1)	(0,0,0)
	h1 + aksi = l2 + h2		a1 a2 ahb2	(loc,f12,l1− aksi + 1,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (1,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2,μ 2) (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2 + 1,μ 2 + 1)	(0,0,0)
			ahb1	(2,r,l1− aksi,0,0,1,1,0,1,1)	(h1 + aksi - μ 1,0,μ 1)
	h1 + aksi > l2 + h2		a1 a2 ahh	(3,ir,l1− saham + 1,0,0,0,0,0,0,0,0) (3,ir,l1− aksi,1,0,0,0,0,0,0) ((3,r, l1− aksi,0,0,1,1,0,1,1))	(h1 + aksi - μ 1,0,μ 1)

	Mengadopsi	f23,f123,tempat = 2	a1c2 a1(1 − c2) a2 αhγ2 αh(1 − c2)	(1,dan,0,0,0,0,0,0,0,0,0) (3,dan,0,0,0,0,0,0,0,0,0) (2,dan,0,0,0,0,0,0,0,0,0) (2,r,0,0,0,0,0,0,0,0,0) (3,r,0,0,0,0,0,0,0,0,0)	(1.h2− μ 2.μ 2) (1.h3− μ 3.μ 3) (0,h2 + 1 - μ 2,μ 2) (0,h2− μ 2,μ 2 + 1) (0.h3− μ 3.μ 3 + 1)
		f23,f123,tempat! = 2	a1c2 a1(1 − c2) a2 αhγ2 αh(1 − c2)	(1,dan,0,0,0,0,0,0,0,0,0) (3,dan,0,0,0,0,0,0,0,0,0) (2,dan,0,0,0,0,0,0,0,0,0) (2,r,0,0,0,0,0,0,0,0,0) (3,r,0,0,0,0,0,0,0,0,0)	(1.h2− μ 2.μ 2) (1 + h3− μ 3,0,μ 3) (0,h2 + 1 - μ 2,μ 2) (0,h2− μ 2,μ 2 + 1) (h3− μ 3,0,μ 3 + 1)
		(r,and,loc = 2) ,f12	a1 a2 ah	(3,dan,1,0,0,0,0,0,0,0) (3,dan,0,1,0,0,0,0,0,0,0) (3,r,0,0,0,1,1,0,1,1)	(0.h2− μ 2.μ 2)
		(r,ir,loc! = 2) ,f13	a1 a2 ah	(3,dan,1,0,0,0,0,0,0,0) (3,dan,0,1,0,0,0,0,0,0,0) (3,r,0,0,0,1,1,0,1,1)	(h3− μ 3.0.μ 3)
l2 + h2 = h3	tindakan = 0	f12	a1 a2 ahb1 ahb2	(loc,fork,l1 + 1,l2,h1,h2,h3,μ 1,μ 2,μ 3) (1,r,l1,l2,h1,h2 + 1,h2 + 1,μ 1,μ 2,μ 2) (2,r,l1,l2,0,1,1,0,1,1) (1,r,l1,l2,h1,h2 + 1,h3 + 1.μ 1.μ 2 + 1.μ 2 + 1)	(0,0,0) (0,0,0) (h1− μ 1.0.μ 1) (0.0.0)

		f13	a1 *a2c1* a2(1 − *c1*) *ahy1* *αh*(1 − *c1*)	(*1oc, garpu, l1* + 1,l2, h1, h2, h3,μ *1.μ 2,μ* 3) (2,r,l1,l2,0,1,1,0,0,0) (*2,r,l1,l2,0,h3* + *1,h3* + *1.μ 1.μ 3,μ* 3) (2,r,l1,0,0,1,1,0,1,1) (*2,r,l1,l2,h1,h3* + *1,h3* + *1.μ 1.μ* 3 + *1.μ* 3 + 1)	(*0,0,0*) (*h1*− *μ 1.0.μ* 1) (0.0.0) (*h1*− *μ 1.0.μ* 1) (0.0.0)
		f23	a1 *a2* *ahc2* *ah*(1 − c2)	(*loc,fork,l1* + 1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*1,r,l1,0,h1,h2* + *1,h2* + *1,μ 1,μ 2,μ* 2) (2,r,l1,0,h1,h2 + *1,h2* + *1.μ 1.μ* 2 + *1.μ* 2 + 1) (2,r,l1,0,h1,h2 + *1,h2* + *1.μ 1.μ* 3 + *1.μ* 3 + 1)	(*0,0,0*)
		f123	a1 *a2* *ahh1* *ahhh2* *ah*(1 − θ1− *ϑ2*)	(*loc,fork,l1* + 1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*2,r,l1,0,h1,h2* + *1,h2* + *1,μ 1,μ 2,μ* 2) (2,r,l1,0,0,1,1,0,1,1) (2,r,l1,0,h1,h3 + *1,h3* + *1.μ 1.μ* 2 + *1.μ* 2 + 1) (2,r,l1,0,h1,h3 + *1,h3* + *1.μ 1.μ* 3 + *1,μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*h1*− *μ* 1.0.μ 1) (*0,0,0*) (*0,0,0*)
		r,dan	a1 *a2* *ahh*	(*loc,ir,l1* + *1,l2,h1,h2,h3,μ 1,μ 2,μ* 3) (*loc,ir,l1,l2* + *1,h1,h2,h3,μ 1,μ 2,μ* 3) (loc,ir,l1,l2,h1,h2 + *1,h3* + *1,μ 1,μ* 2 + *1,μ* 3 + 1)	(*0,0,0*)
	aksi = h3− *h1* > 0	f23	a1 *a2* *ahth1* *ahh2* *ah*(1 − θ1− *ϑ2*)	(*loc,f123,l1*− *action* + *1,l2,h1* + *share,h2,h3,μ 1,μ 2,μ* 3) (*2,r,l1*− *share,0,h1* + *share,h2* + *1,h3* + *1,μ 1,μ 2,μ* 2) (*2,r,l1*− aksi,l2,0,1,1,0,1,1) (*2,r,l1*− *share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 2 + *1,μ* 2 + 1) (*2,r,l1*− *share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 3 + *1,μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*h1*− *μ 1.0.μ* 1) (0.0.0) (*0,0,0*)
		r,h2 = *μ* 2	a1 a2c1 a2(1 − c1) *ahg1* *ah*(1 − c1)	(*loc,f13,l1*− *aksi* + *1,l2,h1* + *share,h2,h3,μ 1,μ 2,μ* 3) (*2,r,l1*− aksi,l2,0,1,1,0,0,0) (*2,r,l1*− *share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 3,μ 3) (*2,r,l1*− aksi,l2,0,1,1,0,1,1) (*2,r,l1*− *share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 3 + *1,μ* 3 + 1)	(*0,0,0*) (*h1* + *saham* - *μ* 1,0,μ 1) (*0,0,0*) (*h1* + *aksi* - *μ 1,0,μ* 1) (*0,0,0*)
		r,h2! = *μ* 2	a1 *a2* *ahb1* *ahb2*	(*loc,f12,l1*− *aksi* + *1,l2,h1* + *share,h2,h3,μ 1,μ 2,μ* 3) (*2,r,l1*− *share,l1,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 2,μ 2) (*2,r,l1*− aksi,l2,0,1,1,0,1,1) (*2,r,l1*− *share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* 3 + *1,μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*h1* + *aksi* - *μ 1,0,μ* 1) (*0,0,0*)
	h1 + *aksi* > *l2* + *h2*		a1 *a2* *ahh*	(*3,ir,l1*− *saham* + 1,0,0,0,0,0,0,0,0) (*3,ir,l1*− aksi,1,0,0,0,0,0,0) ((*3,r, l1*− aksi,0,0,1,1,0,1,1))	(h1 + *aksi* - *μ 1.0.μ* 1)

TABEL IV

DESKRIPSI TRANSISI DAN PENGHARGAAN MATRIKS P DAN R DALAM MASALAH KEPUTUSAN MPO.

l2 + *h2*− *h3* > 2	*Mengadopsi*	*tempat*! = 2	(1 − *p*) *α* 1p *α* 2p *α* *hp*	(1,ir,0,l2,h3,h2,h3,μ *3,μ 2,μ* 3) (1,ir,1,l2,h3,h2,h3,μ *3,μ 2,μ* 3) (*1,ir,0,l2* + *1,h3,h2,h3,μ 3,μ 2,μ* 3) (1,r,0,l2,h3,h2,h3 + *1.μ 3.μ 2.μ* 3 + 1)	(*0,0,0*)

		tempat = 2	(1 – *p*) α 1p α 2p α *hp*	(1,ir,0,l2,h3– h2,0,h3– *h2,h3*– *h2,0,h3*– *h2*) (1,ir,1,l2,h3– h2,0,h3– *h2,h3*– *h2,0,h3*– *h2*) (*1,ir,0,l2* + *1,h3*– h2,0,h3– h2,h3– *h2,0,h3*– *h2*) (1,r,0,l2,h3– h2,0,h3– *h2* + *1,h3*– h2,0,h3– *h2* + 1)	(*0.h2*– μ *2.*μ 2)
	h1 + *aksi* < h3		1 - *p* α *1p* α 2p *α hp*	(*loc,ir,l1*– *action,l2,h1* + action,h2,h3,μ *1,*μ *2,*μ 3) (*loc,ir, l1*– *action* + 1,l2,h1 + share,h2,h3,μ *1,*μ *2,*μ 3) (*loc,ir,l1*– *action,l2* + *1,h1* + share,h2,h3,μ *1,*μ *2,*μ 3) (*loc,r,l1*– *action,l2,h1* + *share,h2,h3* + *1,*μ *1,*μ *2,*μ 3 + 1)	(*0,0,0*)
	h1 + *aksi* = h3	*r, aksi* > 0 f13,action = 0	(1 – *p*) α 1p α *2p* α *hpγ1 α* hp(1 - γ 1)	(*loc,f13,l1*– action,l2,h3,h2,h3,μ *1,*μ *2,*μ 3) (*loc,f13,l1*– *aksi* + 1,l2,h3,h2,h3,μ *1,*μ *2,*μ 3) (*loc,f13,l1*– *action,l2* + *1,h3,h2,h3,*μ *1,*μ *2,*μ 3) (*1,r,l1*– share,l2,h3,h2,h3 + *1,*μ *1,*μ *2,*μ 1 + 1) (*1,r,l1*– share,l2,h3,h2,h3 + *1,*μ *1,*μ *2,*μ 3 + 1)	(*0,0,0*)
	h3 < *h1* + *aksi* < *l2* + *h2*– 1		(1 – *p*) α 1p α *2p* α *hp*	(*1,ir,l1*– *share,l2,h1* + *share,h2,h1* + *share,*μ *1,*μ *2,*μ 1) (*1,ir,l1*– *share* + 1,l2,h1 + *share,h2,h1* + *share,*μ *1,*μ *2,*μ 1) (*1,ir,l1*– *share,l2* + *1,h1* + *share,h2,h1* + *share,*μ *1,*μ *2,*μ 1) (*1,r,l1*– *share,l2,h1* + *share,h2,h1* + *share* + *1.*μ *1.*μ *2.*μ 1 + 1)	(*0,0,0*)
	h1 + *aksi* = *l2* + *h2*– 1		(1 – *p*) α 1p α *2p* α *hp*	(*2,ir,l1*– *share,0,h1* + *share,h2* + l2,h2 + l2,μ *1,*μ *2,*μ 2) (*2,r,l1* + 1 – share,0,h1 + *share,h2* + *l2,h2* + l2,μ *1,*μ *2,*μ 2) (*2,r,l1*– *share,1,h1* + *share,h2* + l2,h2 + l2,μ *1,*μ *2,*μ 2) (*2,r,l1*– *share,0,h1* + *share,h2* + *l2* + *1,h2* + l2 + *1,*μ *1,*μ 2 + *1,*μ 2 + 1)	(*0,0,0*)
	h1 + *aksi* = *l2* + *h2*		1 – *p* α *1p* α 2p *α hpβ1* *ahpb2*	(*loc,f12,l1*– action,0,h1 + *action,h2* + *l2,h2* + l2,μ *1,*μ *2,*μ 3) (*loc,f12,l1*– *aksi* + 1,0,h1 + *share,h2* + l2,h2 + l2,μ *1,*μ *2,*μ 3) (*1,r,l1*– *share,0,h1* + *share,h2* + *l2* + *1,h2* + l2 + *1,*μ *1,*μ *2,*μ 2) (*2,r,l1*– *share,0,h1* + *share,h2* + *l2* + *1,h2* + l2 + *1,*μ *1,*μ 2 + *1,*μ 2 + 1) (*2,r,l1*– aksi,0,0,1,1,0,1,1)	(*0,0,0*) (*h1* + *aksi* - μ *1,0,*μ 1)
	h1 + *aksi* > *l2* + *h2*		(1 – *p*) α 1p α *2p* α *hp*	(*3,ir, l1*– aksi,0,0,0,0,0,0,0) (*3,ir,l1*– *saham* + 1,0,0,0,0,0,0,0,0) (*3,ir,l1*– aksi,1,0,0,0,0,0,0) ((*3,r, l1*– aksi,0,0,1,1,0,1,1))	(h1 + *aksi* - μ *1.0.*μ 1)
l2 + *h2*– *h3* = 2	*Mengadopsi*	*tempat*! = 2	(1 – *p*) α 1p α 2p α *hp*	(1,ir,0,l2,h3,h2,h3,μ 3,μ 2,μ 3) (1,ir,1,l2,h3,h2,h3,μ 3,μ 2,μ 3) (*1,ir,0,l2* + *1,h3,h2,h3,*μ *3,*μ *2,*μ 3) (2,r,0,0,h3,h2 + l2,h2 + l2,μ *3,*μ *2,*μ 2)	(*0,0,0*)
		tempat = 2	(1 – *p*) α 1p α 2p α *hp*	(1,ir,0,l2,h3– h2,0,h3– *h2,h3*– *h2,0,h3*– *h2*) (1,ir,1,l2,h3– h2,0,h3– *h2,h3*– *h2,0,h3*– *h2*) (*1,ir,0,l2* + *1,h3*– h2,0,h3– *h2,h3*– h2,0,h3– h2) (*2,r,0,0,h3* - h2,l2,l2,h3– *h2,0,0*)	(*0.h2*– μ *2.*μ 2)
	h1 + *aksi* < h3		(1 – *p*) α 1p α 2p α *hp*	(*loc,ir,l1*– *action,l2,h1* + action,h2,h3,μ *1,*μ *2,*μ 3) (*loc,ir, l1*– *action* + 1,l2,h1 + share,h2,h3,μ *1,*μ *2,*μ 3) (*loc,ir,l1*– *action,l2* + *1,h1* + share,h2,h3,μ *1,*μ *2,*μ 3) (*2,r,l1*– *share,0,h1* + *share,h2* + l2,h2 + l2,μ *1,*μ *2,*μ 2)	(*0,0,0*)
	h1 + *aksi* = h3	*r,aksi* > 0 f13,action = 0	(1 – *p*) α 1p α 2p α *hp*	(*loc,f13,l1*– action,l2,h3,h2,h3,μ *1,*μ *2,*μ 3) (*loc,f13,l1*– *aksi* + 1,l2,h3,h2,h3,μ *1,*μ *2,*μ 3) (*loc,f13,l1*– *action,l2* + *1,h3,h2,h3,*μ *1,*μ *2,*μ 3)	(*0,0,0*)

				(2,r,l1− share,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 2)	
	$_{h1}$ + aksi = l2 + h2− 1		(1 − p) α $_{1p}$ α 2p α $_{hp}$	(2,ir,l1− share,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 2) (2,r,l1 + 1 − share,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 2) (2,r,l1− share,1,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 2) (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 1,μ $_2$ + 1,μ $_2$ + 1)	(0,0,0)
	$_{h1}$ + aksi = l2 + h2		(1 − p) α $_{1p}$ α 2p ahpb2	(loc,f12,l1− action,0,h1 + action,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (loc,f12,l1− aksi + 1,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (1,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2,μ 2) (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 1,μ $_2$ + 1,μ $_2$ + 1)	(0,0,0)
			ahpb1	(2,r,l1− aksi,0,0,1,1,0,1,1)	(h1 + aksi - μ 1,0,μ $_1$)
	$_{h1}$ + aksi > l2 + h2		(1 − p) α $_{1p}$ α 2p α $_{hp}$	(3,ir, l1− aksi,0,0,0,0,0,0,0) (3,ir,l1− saham + 1,0,0,0,0,0,0,0,0) (3,ir,l1− aksi,1,0,0,0,0,0,0) ((3,r, l1− aksi,0,0,1,1,0,1,1))	(h1 + aksi - μ 1,0,μ $_1$)
l2 + h2− h3 = 1	Mengadopsi	tempat = 2,tempat = 3	(1 − p) α $_{1p}$ α $_{2p}$ α $_{hp}$	(1,ir,0,l2,h3− $_{h2,0,h3}$− h2,h3− h2,0,h3− h2) (1,ir,1,l2,h3− $_{h2,0,h3}$− h2,h3− h2,0,h3− h2) (1,ir,0,l2 + 1,h3− h2,0,h3− h2,h3− h2,0,h3− $_{h2}$) (loc,f23,0,0 ,$_{h3}$− h2,h3− h2 + 1,h3− $_{h2}$ + 1,0,0,1)	(0.h2− μ 2.μ $_2$)
	$_{h1}$ + aksi < $_{h3}$		(1 − p) α $_{1p}$ α $_{2p}$ α $_{hp}$	(loc,ir,l1− action,l2,h1 + action,h2,h3,μ 1,μ 2,μ 3) (loc,ir, l1− action + 1,l2,h1 + share,h2,h3,μ 1,μ 2,μ 3) (loc,ir,l1− action,l2 + 1,h1 + share,h2,h3,μ 1,μ 2,μ 3) (loc,f23,l1− action,0,h1 + share,h3 + 1,h3 + 1,μ 1,μ 2,μ 3 + 1)	(0,0,0)
	$_{h1}$ + aksi = l2 + h2		(1 − p) α $_{1p}$ α 2p ahpb2	(loc,f12,l1− action,0,h1 + action,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (loc,f12,l1− aksi + 1,0,h1 + share,h2 + l2,h2 + l2,μ 1,μ 2,μ 3) (1,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 2,μ 2) (2,r,l1− share,0,h1 + share,h2 + l2 + 1,h2 + l2 + 1,μ 1,μ 1,μ $_2$ + 1,μ $_2$ + 1)	(0,0,0)
			ahpb1	(2,r,l1− aksi,0,0,1,1,0,1,1)	(h1 + aksi - μ 1,0,μ $_1$)
	$_{h1}$ + aksi > l2 + h2		(1 − p) α $_{1p}$ α 2p α $_{hp}$	(3,ir, l1− aksi,0,0,0,0,0,0,0) (3,ir,l1− saham + 1,0,0,0,0,0,0,0,0) (3,ir,l1− aksi,1,0,0,0,0,0,0) ((3,r, l1− aksi,0,0,1,1,0,1,1))	(h1 + aksi - μ 1,0,μ $_1$)

l2 + $_{h2}$ = h3	Mengadopsi	f23,f123,tempat = 2	(1 − p) α 1pγ2 α $_{1p}$(1 − γ $_2$) α $_{2p}$ α $_{hp}$γ2 α $_{hp}$(1 - γ $_2$)	(loc,f23,0,l2,0,μ $_3$− μ 2.μ $_3$− μ 2.0.μ $_3$− μ 2.μ 3− μ $_2$) (1,dan,0,0,0,0,0,0,0,0,0) (3,dan,0,0,0,0,0,0,0,0) (2,dan,0,0,0,0,0,0,0,0) (2,r,0,0,0,0,0,0,0,0) (3,r,0,0,0,0,0,0,0,0)	(0.h3− μ 3.μ $_2$) (1.h2− μ 2.μ $_2$) (1.h3− μ 3.μ $_3$) (0,h2 + 1 - μ 2,μ $_2$) (0,h2− μ 2,μ $_2$ + 1) (0.h3− μ 3.μ $_3$ + 1)
		f23,f123,tempat ! = 2	(1 − p) α 1pγ2 α $_{1p}$(1 − γ $_2$) α $_{2p}$	(loc,f23,0,l2,0,h2,h3,μ 1,μ 2,μ 3) (1,dan,0,0,0,0,0,0,0,0,0) (3,dan,0,0,0,0,0,0,0,0) (2,dan,0,0,0,0,0,0,0,0)	(0,0,0) (1.h2− μ 2.μ $_2$) (1 + h3− μ 3,0,μ 3) (0.h2 + 1 - μ 2.μ $_2$)

		α *hpγ2 α* $_{hp}$(1 - γ $_{2}$)	(2,r,0,0,0,0,0,0,0,0) (3,r,0,0,0,0,0,0,0,0)	(*0,h2− μ 2,1* + μ $_{2}$) (*h3− μ* 3.0.μ $_{3}$ + 1)
	(*r,and,loc* = 2) $_{,f12}$	(1 − *p*) α $_{1p}$ α $_{2p}$ α $_{hp}$	(3,dan,0,0,0,0,0,0,0,0) (*3,dan,1,0,0,0,0,0,0,0)* (3,dan,0,1,0,0,0,0,0,0) (3,r,0,0,0,1,1,0,1,1)	(*0.h2− μ 2.μ* $_{2}$)
	(*r,ir,loc*! = 2) $_{,f13}$	(1 − *p*) α $_{1p}$ α $_{2p}$ α $_{hp}$	(3,dan,0,0,0,0,0,0,0,0) (*3,dan,1,0,0,0,0,0,0,0)* (3,dan,0,1,0,0,0,0,0,0) (3,r,0,0,0,1,1,0,1,1)	(*h3− μ* 3.0.μ $_{3}$)
tindakan = 0	f12	(1 − *p*) α *1p α* $_{2p}$ α *hpϐ1 α* *hpϐ2*	(loc,fork,l1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*loc,fork,l1* + 1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*1,r,l1,l2,h1,h2* + *1,h2* + *1,μ 1,μ 2,μ* $_{2}$) (2,r,l1,l2,0,1,1,0,1,1) (*1,r,l1,l2,h1,h2* + *1,h3* + *1.μ 1.μ* $_{2}$ + *1.μ* 2 + 1)	(*0,0,0*) (*0,0,0*) (*0,0,0*) (*h1− μ 1.0.μ* $_{1}$) (0.0.0)
	f13	(1 − *p*) α $_{1p}$ *α2pc1* α $_{2p}$(1 − γ $_{1}$) α *hpγ1* α $_{hp}$(1 − γ $_{1}$)	(1oc, garpu, l1, l2, h1, h2, h3,μ *1.μ 2,μ* 3) (*1oc, garpu, l1* + 1,l2, h1, h2, h3,μ *1.μ 2,μ* $_{3}$) (2,r,l1,l2,0,1,1,0,0,0) (*2,r,l1,l2,0,h3* + *1,h3* + *1.μ 1.μ 3,μ* $_{3}$) (2,r,l1,0,0,1,1,0,1,1) (*2,r,l1,l2,h1,h3* + *1,h3* + *1.μ 1.μ* $_{3}$ + *1.μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*h1− μ 1.0.μ* $_{1}$) (0.0.0) (*h1− μ 1.0.μ* $_{1}$) (0.0.0)
	f23	(1 − *p*) α $_{1p}$ α *2p* *ahpg2* α $_{hp}$(1 − γ $_{2}$)	(loc,fork,l1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*loc,fork,l1* + 1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*1,r,l1,0,h1,h2* + *1,h2* + *1,μ 1,μ 2,μ* $_{2}$) (2,r,l1,0,h1,h2 + *1,h2* + *1.μ 1.μ* $_{2}$ + *1.μ* 2 + 1) (2,r,l1,0,h1,h2 + *1,h2* + *1.μ 1.μ* $_{3}$ + *1.μ* 3 + 1)	(*0,0,0*)
	f123	(1 − *p*) *α 1p α* $_{2p}$ α *hpϑ1* *ahpth2* α $_{hp}$(1 − *ϑ1− ϑ2*)	(loc,fork,l1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*loc,fork,l1* + 1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*2,r,l1,0,h1,h2* + *1,h2* + *1,μ 1.μ 2,μ* 2) (*2,r,l1,0,0,1,1,0,1,1)*) (2,r,l1,0,h1,h3 + *1,h3* + *1.μ 1.μ* $_{2}$ + *1.μ* 2 + 1) (2,r,l1,0,h1,h3 + *1,h3* + *1.μ 1.μ* $_{3}$ + *1,μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*0,0,0*) (*h1− μ* 1.0.μ $_{1}$) (*0,0,0*) (*0,0,0*)
	r,dan	(1 − *p*) α $_{1p}$ α *2p* α $_{hp}$	(loc,ir,l1,l2,h1,h2,h3,μ *1,μ 2,μ* 3) (*loc,ir,l1* + *1,l2,h1,h2,h3,μ 1,μ 2,μ* $_{3}$) (*loc,ir,l1,l2* + *1,h1,h2,h3,μ 1,μ 2,μ* $_{3}$) (loc,ir,l1,l2,h1,h2 + *1,h3* + *1,μ 1,μ* 2 + *1,μ* $_{3}$ + 1)	(*0,0,0*)
aksi = $_{h3}$− *h1* > 0	f23	(1 − *p*) α *1p α* $_{2p}$ α *hpϑ1 α* *hpϑ2* α $_{hp}$(1 − *ϑ1− ϑ2*)	(*loc,f123,l1− action,l2,h1* + *action,h2,h3,μ 1,μ 2,μ* 3) (*loc,f123,l1− action* + *1,l2,h1* + *share,h2,h3,μ 1,μ 2,μ* 3) (*2,r,l1− share,0,h1* + *share,h2* + *1,h3* + *1,μ 1,μ 2,μ* 2) (*2,r,l1−* aksi,l2,0,1,1,0,1,1) (*2,r,l1− share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* $_{2}$ + *1,μ* 2 + 1) (*2,r,l1− share,l2,h1* + *share,h2* + *1,h3* + *1,μ 1,μ* $_{3}$ + *1,μ* 3 + 1)	(*0,0,0*) (*0,0,0*) (*0,0,0*) (*h1− μ 1.0.μ* $_{1}$) (0.0.0) (*0,0,0*)

		$_{r,h2} = \mu_{2}$	$(1-p)\ \alpha_{1p}$ $\alpha 2p\gamma 1_{\alpha 2p}(1-c1)$ $ahpg1$ $\alpha_{hp}(1-\gamma_{1})$	$(loc,f13,l1-action,l2,h1+action,h2,h3,\mu_{1},\mu_{2},\mu_{3})$ $(loc,f13,l1-aksi+1,l2,h1+share,h2,h3,\mu_{1},\mu_{2},\mu_{3})$ $(2,r,l1-$ aksi,l2,0,1,1,0,0,0 $)$ $(2,r,l1-share,l2,h1+share,h2+1,h3+1,\mu_{1}\ _{3,\mu_{3}})$ $(2,r,l1-$ aksi,l2,0,1,1,0,1,1 $)$ $(2,r,l1-share,l2,h1+share,h2+1,h3+1,\mu_{1}\ _{3}+1,\mu_{3}+1)$	$(0,0,0)$ $(0,0,0)$ $(h1+aksi-\mu_{1}.0.\mu_{1})$ $(0.0.0)$ $(h1+aksi-\mu_{1}0,\mu_{1})$ $(0,0,0)$
		$_{r,h2}! = \mu_{2}$	$(1-p)\ \alpha_{1p}$ α_{2p} $ahpb1$ $ahpb2$	$(loc,f12,l1-action,l2,h1+action,h2,h3,\mu_{1},\mu_{2},\mu_{3})$ $(loc,f12,l1-aksi+1,l2,h1+share,h2,h3,\mu_{1},\mu_{2},\mu_{3})$ $(2,r,l1-share,l1,h1+share,h2+1,h3+1,\mu_{1}\ _{2,\mu_{2}})$ $(2,r,l1-$ aksi,l2,0,1,1,0,1,1 $)$ $(2,r,l1-share,l2,h1+share,h2+1,h3+1,\mu_{1}\ _{3}+1,\mu_{3}+1)$	$(0,0,0)$ $(0,0,0)$ $(0,0,0)$ $(h1+aksi-\mu_{1}0,\mu_{1})$ $(0,0,0)$
	$_{h1}+aksi>l2+h2$		$(1-p)$ $\alpha_{1p}\ \alpha 2p$ α_{hp}	$(3,ir,l1-$ aksi,0,0,0,0,0,0,0,0 $)$ $(3,ir,l1-saham+$ 1,0,0,0,0,0,0,0,0,0 $)$ $(3,ir,l1-$ aksi,1,0,0,0,0,0,0 $)$ $((3,r,l1-$ aksi,0,0,1,1,0,1,1 $))$	$(h1+aksi-\mu_{1}0,\mu_{1})$