

PENYELESAIAN PROBLEMA TOWER OF HANOI MENGGUNAKAN ALGORITMA A*

Supiyandi

Fakultas Ilmu Komputer Program Studi Sistem Informasi
Universitas Potensi Utama

Jl. KL. Yos Sudarso KM 6.5 No. 3A Tanjung Mulia, Medan, North Sumatera, 20241, Indonesia

Email :karyaprima.lkp@gmail.com

Abstrak

*Puzzle atau biasa disebut dengan permainan ketangkasan, puzzle sudah ada sejak zaman dahulu dan biasanya berupa kepingan ataupun potongan batu yang harus disusun sehingga membentuk suatu objek, puzzle mencapai puncaknya pada tahun 1900an dimana puzzle berupa kubik menjadi primadona pada saat itu dan hingga sekarang masih menjadi primadona, banyaknya puzzle yang ada membutuhkan penyelesaian yang berbeda termasuk salah satunya adalah tower of hanoi. Menara Hanoi yang juga disebut Menara Brahma merupakan sebuah permainan matematis atau teka-teki. Teka-teki ini terdiri dari tiga tiang dan sejumlah cakram dengan ukuran yang berbeda yang bisa dimasukkan ke tiang mana saja. Permainan dimulai dengan cakram-cakram yang tertumpuk rapi berurutan berdasarkan ukurannya dalam salah satu tiang, cakram terkecil diletakkan teratas, sehingga membentuk kerucut. Tujuan dari permainan matematis ini adalah memindahkan seluruh cakram dari satu tiang ke tiang yang lain dengan beberapa aturan. Dalam permasalahan menara hanoi ini, solusi berusaha didapatkan dengan algoritma A**

Kata kunci : *Puzzle, Algoritma A*, Hanoi Tower, Puzzle Hanoi*

1. PENDAHULUAN

Puzzle atau biasa disebut dengan permainan ketangkasan, puzzle sudah ada sejak zaman dahulu dan biasanya berupa kepingan ataupun potongan batu yang harus disusun sehingga membentuk suatu objek, puzzle mencapai puncaknya pada tahun 1900an dimana puzzle berupa kubik menjadi primadona pada saat itu dan hingga sekarang masih menjadi primadona, banyaknya puzzle yang ada membutuhkan penyelesaian yang berbeda termasuk salah satunya adalah *tower of hanoi*.

Menara Hanoi adalah sebuah permainan matematis atau teka-teki. Teka-teki ini ditemukan Eduard Lucas, ahli matematika Perancis di tahun 1883, metode penyelesaian menara hanoi ini sangat banyak beberapa diantaranya adalah algoritma BFS, DFS dan bidirectional A+ dan yang terakhir adalah algoritma A* yang sedang diteliti penulis.

Menara Hanoi yang juga disebut Menara Brahma merupakan sebuah permainan matematis atau teka-teki. Teka-teki ini terdiri dari tiga tiang dan sejumlah cakram dengan ukuran yang berbeda yang bisa dimasukkan ke tiang mana saja. Permainan dimulai dengan cakram-cakram yang tertumpuk rapi berurutan berdasarkan ukurannya dalam salah satu tiang, cakram terkecil diletakkan teratas, sehingga membentuk kerucut. Tujuan dari permainan matematis ini adalah memindahkan seluruh cakram dari satu tiang ke tiang yang lain dengan beberapa aturan. Dalam permasalahan menara hanoi ini, solusi berusaha didapatkan dengan algoritma A*.

A* mendeskripsikan proses pemecahan masalah dimana seseorang harus mengambil setiap keputusan terbaik pada setiap langkahnya. Optimisasi dengan dynamic programming yang diterapkan di A* dicapai dengan memilih setiap keputusan terbaik (optimal)

dalam setiap tahap. Dalam penyelesaian Menara Hanoi terdapat beberapa algoritma yang bisa digunakan, termasuk salah satunya adalah Pemrograman Dinamis (*Dynamic Programming*).

Langkah yang digunakan untuk menyelesaikan dengan Berdasarkan uraian di atas, penulis ingin merancang suatu perangkat lunak yang mampu untuk melakukan penyelesaian terhadap Puzzle Tower Hanoi. Oleh karena itu, penulis mengambil penelitian dengan judul "**Penyelesaian Problema Tower Of Hanoi Menggunakan Algoritma A***".

Pada penelitian ini penulis mengambil rumusan masalah dari penelitian ini adalah :

- Bagaimana penerapan algoritma A* dalam menyelesaikan permasalahan tower of hanoi?
 - Bagaimana membuat sebuah aplikasi penyelesaian masalah tower of hanoi dengan menerapkan algoritma A* didalam sebuah model penyelesaian.
- Setelah menentukan rumusan masalah penulis juga menentukan batasan masalah pada penelitian ini adalah sebagai berikut:
- Jumlah cakram dibatasi maksimal 5 buah dan minimal 3 buah.
 - Setiap cakram mempunyai besar yang berbeda, semakin kebawah cakraanya maka akan semakin besar.
 - Jumlah tiang(*tower*) sebanyak 3 buah.
 - Input* dari perangkat lunak berupa keadaan awal (*initial state*) berupa posisi awal setiap cakram dalam tumpukan tertentu.
 - Keadaan akhir (*goal state*) berupa tumpukan cakram yang sama dengan ketentuan cakram awal.

2. METODOLOGI

Metode merupakan suatu cara atau teknik yang sistematis untuk memecahkan suatu kasus sehingga memberikan hasil sesuai dengan yang diharapkan. Penulis menggunakan studi kepustakaan (*library research*), yaitu menggunakan sumber-sumber melalui buku, jurnal, tugas akhir, tesis maupun disertasi, *browsing* melalui internet, serta sumber-sumber lain yang relevan untuk digunakan dalam penelitian ini. Studi kepustakaan dalam penelitian ini adalah hal-hal yang berkaitan dengan permasalahan problema keadaan dan ruang dengan penyelesaian algoritma A*

2.1. KECERDASAN BUATAN

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang kersern dengan pengotomatisasi tingkah laku cerdas. Pernyataan tersebut juga dapat dijadikan definisi dari AI. Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada *sound theoretical* (teori suara) dan prinsip-prinsip aplikasi dari bidangnya. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya [1].

Namun, definisi di atas belum memadai karena pada kenyataannya, istilah kecerdasan itu sendiri tidak didefinisikan atau dipahami dengan sangat baik. Sering kali kita meyakini telah mengetahui tingkah laku cerdas saat melihatnya sendiri. Akan tetapi, keraguan juga akan muncul ketika orang lain menyatakan kecerdasan dalam cara khusus menurut cara mereka yang cukup membantu dalam evaluasi kecerdasan program komputer. Dengan demikian, belum ada suatu definisi yang cocok untuk AI itu sendiri. Definisi AI dapat dibagi dalam empat kategori, yaitu:

- Sistem yang dapat berpikir seperti manusia "*Thinking humanly*"
- Sistem yang dapat bertingkah laku seperti manusia "*Acting humanly*"
- Sistem yang dapat berpikir secara rasional "*Thinking rationally*"
- Sistem yang dapat bertingkah laku secara rasional "*Acting rationally*"

Secara historis, keempat pendekatan tersebut telah dilakukan. Pendekatan manusia haruslah merupakan suatu ilmu empiris, termasuk hipotesa dan konfirmasi percobaan dan pendekatan rasional meliputi kombinasi dari matematika dan rekayasa [1].

Ada dua hal yang sangat mendasar mengenai penelitian-penelitian AI, yaitu *knowledge representation* (representasi pengetahuan) dan *search* (pelacakan). Para peneliti AI terus mengembangkan berbagai jenis teknik baru dalam menangani sejumlah permasalahan yang tergolong ke dalam AI seperti

vision dan percakapan, pemrosesan bahasa alami dan permasalahan khusus seperti diagnosa medis [1].

AI seperti bidang ilmu lainnya juga memiliki sejumlah subdisiplin ilmu yang sering digunakan untuk pendekatan yang esensial bagi penyelesaian suatu masalah dan dengan aplikasi bidang AI yang berbeda [1].

2.2. TEKNIK PENCARIN

Pencarian atau pelacakan merupakan salah satu teknik untuk menyelesaikan permasalahan AI. Keberhasilan suatu sistem, salah satunya ditentukan oleh kesuksesan dalam pencarian dan pencocokan. Teknik dasar pencarian memberikan suatu kunci bagi banyak sejarah penyelesaian yang penting dalam bidang AI [2]. Ada beberapa aplikasi yang menggunakan teknik pencarian ini, yaitu:

- Papan *game* dan *puzzle* (tic-tac-toe, catur, block world, tower hanoi)
- Penjadwalan dan masalah *routing* (*travelling salesman problem*)
- Parsing* bahasa dan interpretasinya (pencarian struktur dan arti)
- Logika pemograman (pencarian fakta dan implikasinya)
- Computer vision* dan pengenalan pola
- Sistem pakar berbasis kaidah (*rule based expert system*)

Konsep pencarian untuk suatu solusi dalam ruang keadaan (*state space*) merupakan pusat AI yang menjadikan AI lebih unggul dalam bidang ilmu komputer dibandingkan dengan yang lainnya, dan prinsip kontribusi AI untuk ilmu pengetahuan dari pencarian ini merupakan konsep basis pengetahuan (*knowledge based*) heuristik untuk pembatasan dan pencarian berarah (*directing search*). Pada dasarnya, ada dua teknik pencarian dan pelacakan yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*) [3].

2.3. Algoritma A*

Pencarian buta (*uninformed* atau *blind search*) umumnya kurang efisien. Hal ini disebabkan oleh waktu akses yang cukup lama dan besarnya memori yang diperlukan. Untuk mengatasi kelemahan ini maka penambahan domain pengetahuan akan menghasilkan suatu proses pencarian dan investigasi baru. Pencarian tersebut biasanya diistilahkan dengan *informed search* [4].

George Poyla mendefinisikan A* sebagai studi metode dan kaidah penemuan. Dalam pencarian ruang keadaan, A* dinyatakan sebagai aturan untuk melakukan pemilihan cabang-cabang dalam ruang keadaan yang paling tepat untuk mencapai solusi permasalahan yang dapat diterima [4].

Pemecahan masalah dalam AI menggunakan heuristik dalam dua situasi dasar, yaitu:

- Permasalahan yang mungkin tidak mempunyai solusi yang pasti disebabkan oleh ambiguitas (keraguan / ketidakpastian) mendasar dalam

pernyataan permasalahan atau data yang tersedia. Diagnosa kedokteran merupakan salah satu contohnya, dimana sejumlah gejala mungkin dapat ditimbulkan oleh berbagai macam penyebab yang mungkin. Dokter menggunakan heuristik A^* untuk memilih atau menentukan diagnosa yang paling dapat diharapkan dan memutuskan rencana penanganannya. *Vision* merupakan masalah lainnya untuk permasalahan ketidakpastian yang mendasar.

- b. Permasalahan yang boleh jadi memiliki solusi pasti, tetapi biaya komputasinya untuk mendapatkan solusi tersebut mungkin sangat tinggi. Dalam banyak masalah seperti catur, ruang keadaan bertambah secara luar biasa seiring dengan pertambahan jumlah keadaan yang dimungkinkan.

Seperti semua kaidah penemuan lainnya, *heuristik A** juga dapat salah. *Heuristik A** hanyalah panduan informasi untuk menebak langkah berikutnya yang harus diambil dalam menyelesaikan suatu permasalahan, dan sering dilakukan berdasarkan eksperimen/percobaan atau secara intuisi. Oleh karena, menggunakan informasi yang terbatas, *heuristik A** jarang dapat memprediksi tingkah laku yang eksak dari ruang keadaan saat dilakukan pencarian. *Heuristik A** dapat membimbing algoritma pencarian untuk mendapatkan solusi suboptimal atau gagal menemukan solusi apapun [4].

*Heuristik A** dan perancangan algoritma untuk mengimplementasikan pencarian heuristik telah menjadi inti permasalahan penelitian I. *Game playing* dan pemecahan teorema (*theorem solving*) adalah dua aplikasi paling tua dari AI, kedua-duanya memerlukan heuristik untuk memangkaskan ruang dari solusi yang mungkin.

Saat ini, penelitian mengenai sistem pakar telah menyadari arti penting *heuristik A** sebagai komponen yang esensial dalam pemecahan permasalahan. Bila seorang pakar memecahkan suatu permasalahan maka dia akan menguji informasi yang tersedia dan membuat suatu keputusan [4].

2.4. Tower of Hanoi

Menara Hanoi adalah sebuah permainan matematis atau teka-teki. Teka-teki ini ditemukan Eduard Lucas, ahli matematika Perancis di tahun 1883.



Gambar 1 Kondisi Awal Menara Hanoi

Permainan ini terdiri dari tiga tiang dan sejumlah cakram dengan ukuran berbeda-beda yang

bias dimasukkan ke tiang mana saja. Permainan Menara Hanoi dimulai dengan cakram-cakram yang tertumpuk rapi dari cakram paling besar sampai ke cakram paling terkecil dalam salah satu tiang, sehingga membentuk kerucut.

Objektif dari permainan Menara Hanoi adalah memindahkan tumpukan n buah cakram berlubang dari tiang asal ke tiang tujuan dengan memanfaatkan sebuah tiang perantara. Piringan berukuran tidak sama. Jumlah pemindahan dalam n buah cakram adalah sebanyak $2^n - 1$ kali.

Permainan Menara Hanoi memiliki beberapa aturan yang harus dipatuhi untuk menyelesaikan teka-teki dalam melakukan pemindahan cakram harus mengikuti aturan berikut:

- Hanya satu cakram yang boleh dipindahkan dalam setiap kali perpindahan.
- Setiap perpindahan berupa pengambilan cakram teratas dari satu tiang dan memasukkannya ke tiang lain, di atas cakram lain yang mungkin sudah ada di tiang tersebut.
- Tidak boleh meletakkan cakram di atas cakram lain yang lebih kecil.

Tower of Hanoi sering digunakan dalam riset psikologi dalam pemecahan masalah. Terdapat pula variasi dari *Tower of Hanoi*, yaitu *Tower of London* untuk diagnose neuropsikologi dan pengerjaan fungsi eksklusif. *Tower of Hanoi* juga sering dipakai dalam skema *Backup Rotation* ketika membuat penggandaan data komputer dimana banyak tape/media penyimpanan termasuk didalamnya. *Tower of Hanoi* dapat dipakai untuk melatih kreativitas anak-anak dalam masa pertumbuhan. Selain itu, *Tower of Hanoi* juga sering diimplementasikan dalam proses pengajaran algoritma rekursif dasar. *Tower of Hanoi* juga digunakan untuk tes memory oleh para neuropsikolog untuk mengevaluasi amnesia [5].

3. ANALISA DAN HASIL

Puzzle Tower Hanoi merupakan salah satu persoalan klasik dalam bidang studi *Artificial Intelligence* (AI). Problema ini dapat diilustrasikan seperti berikut, terdapat 3 atau lebih cakram yang disusun sebagai kondisi awal (*initial state*). Sasaran (*goal*) dari kasus ini adalah mendapatkan suatu tumpukan cakram yang sesuai dengan kondisi awal. Operasi, aksi dan aturan yang terdapat di dalam kasus ini adalah:

- Kondisi awal dari menara hanoi, terdapat 3 buah cakram yang diberi nama C_1, C_2, C_3 , C_1 merupakan cakram yang terbesar dan C_3 merupakan cakram yang paling kecil, berikut gambarnya



Gambar 2 Kondisi Awal

2. Pindahkan cakram merah dari tiang-1 ke tiang-3. Aturannya adalah cakram harus merupakan balok paling atas dari tiang-1 dan cakram merah menempati posisi paling atas pada tiang-3.



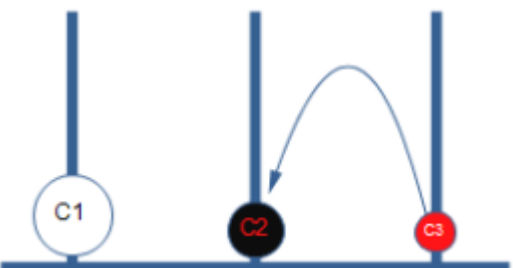
Gambar 3 Operasi-1 Pemindahan Cakram dari tiang-1 ke tiang-3

3. Pindahkan cakram hitam dari tiang-1 ke tiang-2. Aturannya adalah cakram harus merupakan cakram paling atas dari tiang-1 dan cakram hitam menempati posisi paling atas pada tiang-2.



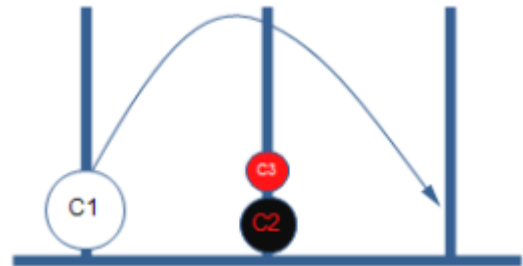
Gambar 4 Operasi-2 Pemindahan Cakram dari tiang-1 ke tiang-2

4. Pindahkan cakram merah dari tiang-3 ke tiang-2. Aturannya adalah cakram merah harus merupakan cakram paling atas dari tiang-3 dan cakram merah menempati posisi paling atas pada tiang-2.



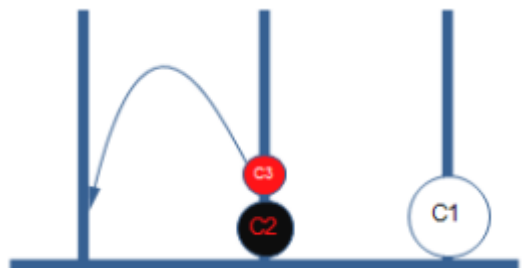
Gambar 5 Operasi-3 Pemindahan Cakram dari tiang-3 ke tiang-2

5. Pindahkan cakram putih dari tiang-1 ke tiang-3. Aturannya adalah cakram putih harus merupakan cakram paling atas dari tiang-1 dan cakram putih menempati posisi paling atas pada tiang-3.



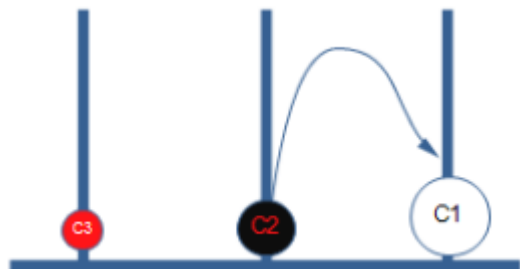
Gambar 6 Operasi-5 atau Aksi-5

6. Pindahkan cakram merah dari tiang-2 ke tiang-1. Aturannya adalah cakram merah harus merupakan cakram paling atas dari tiang-2 dan cakram merah menempati posisi paling atas pada tiang-1.



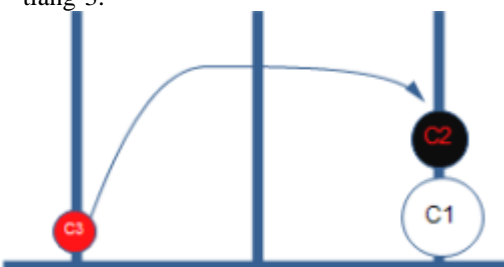
Gambar 7 Operasi-6 atau Aksi-6

7. Pindahkan cakram hitam dari tiang-2 ke tiang-3. Aturannya adalah cakram hitam harus merupakan cakram paling atas dari tiang-2 dan cakram hitam menempati posisi paling atas pada tiang-3.



Gambar 8 Operasi-7 atau Aksi-7

8. Pindahkan cakram merah dari tiang-1 ke tiang-3. Aturannya adalah cakram merah harus merupakan cakram paling atas dari tiang-1 dan cakram merah menempati posisi paling atas pada tiang-3.

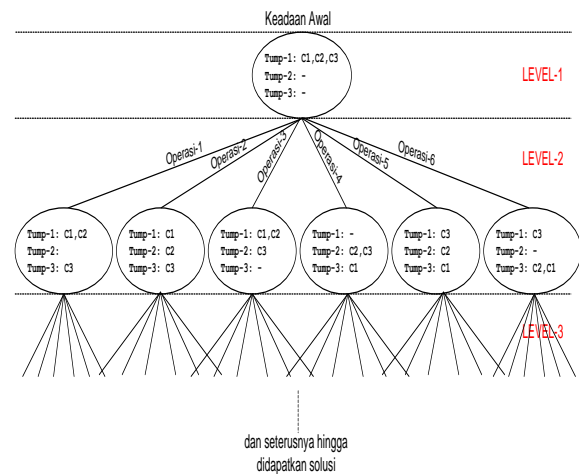


Gambar 9 Operasi-8 atau Aksi-8

Prosedur pencarian A^* merupakan pencarian yang dilakukan dengan mengunjungi tiap-tiap *node* secara sistematis pada setiap level hingga keadaan tujuan (*goal state*) ditemukan. Atau dengan kata lain, penelusuran dilakukan dengan mengunjungi *node-node* per level hingga ditemukan *goal state*. Apabila terdapat solusi, pencarian A^* menjamin ditemukannya solusi dengan lintasan terpendek (*shortest path*). Penerapan algoritma A^* dapat dilihat pada contoh berikut.

1. Keadaan awal I:
 - i. Tumpukan-1: C1, C2,C3 (Balok C3 berada di atas balok C2 dan C1).
 - ii. Tumpukan-2: -
 - iii. Tumpukan-3: -
2. Keadaan awal II:
 - i. Tumpukan-1: C1, C2(Balok C2 berada di atas balok C1)
 - ii. Tumpukan-2: -
 - iii. Tumpukan-3: C3
3. Keadaan awal III:
 - i. Tumpukan-1: C1.
 - ii. Tumpukan-2: C2.
 - iii. Tumpukan-3: C3.
4. Keadaan awal IV:
 - i. Tumpukan-1: C1.
 - ii. Tumpukan-2: C2,C3(Balok C3 berada di atas balok C2)
 - iii. Tumpukan-3: -
5. Keadaan awal V:
 - i. Tumpukan-1: -.
 - ii. Tumpukan-2: C2,C3(Balok C3 berada di atas balok C2)
 - iii. Tumpukan-3: C1.
6. Keadaan awal VI:
 - i. Tumpukan-1: C3.
 - ii. Tumpukan-2: C2.
 - iii. Tumpukan-3: C1.
7. Keadaan awal VII:
 - i. Tumpukan-1: C3.
 - ii. Tumpukan-2: .
 - iii. Tumpukan-3: C2,C1(Balok C2 berada di atas balok C1)
8. Keadaan akhir:
 - i. Tumpukan-1: -.
 - ii. Tumpukan-2: -.
 - iii. Tumpukan-3: C1, C2,C3. (Balok C3 berada di atas balok C2 dan C1).

Penelusuran A^* seperti terlihat pada gambar 10.



Gambar 10 Penelusuran A^*

Pada gambar 10, terlihat bahwa pencarian dimulai dari *node* akar yang berisi problema. Pada *node* akar dilakukan beberapa operasi sehingga didapatkan keadaan-keadaan baru. Keadaan-keadaan baru ini direpresentasikan sebagai *node* anak dari *node* akar. Selanjutnya hal yang sama dilakukan untuk *node-node* yang baru terbentuk hingga ditemukan solusi.

4. KESIMPULAN DAN SARAN

Kesimpulan dan saran dari penelitian ini, Setelah menyelesaikan perangkat lunak penyelesaian probleme Tower Of Hanoi, penulis menarik kesimpulan bawah penggunaan algoritma A^* menjamin solusi pasti ditemukan dengan n langkah, setelah melakukan pembahasan mengenai penelitian ini, algoritma A^* dapat dikembangkan dengan menambahkan beberapa bentuk wadah lainnya, seperti: segi enam (heksagon), segi tujuh (heptagon), segi delapan (octagon) dengan dikombinasikan metode pencarian *Depth-First-Search*, *Hill-Climbing*

DAFTAR PUSTAKA

- [1] T. Sutojo, S.Si.,M.Kom, Kecerdasan Buatan, Penerbit Andi, 2011.
- [2] Sri Kusumadewi, Artificial Intelligence; Teknik dan Aplikasinya, Graha Ilmu, 2003
- [3] William John Teahan, Artificial Intelligence – Agents and Environments, bookboon
- [4] Yulikus Partono, S.Kom, Pengantar Logika dan Algoritma, Penerbit Andi, 2008
- [5] Yaoda Xi, et al, H.M. Revisit The Tower of Hanoi Puzzle, 2001