可変参照型緩対称性推論のモンテカルロ木探索での効果

西村友伸[†1]　大用庫智[†1]　高橋達二[†2]

本研究では甲野により提案された可変参照型緩対称推論をモンテカルロ木探索に応用させ，その効果を測る為にリバーシの AI に実装し，モンテカルロ木探索で広く利用されている UCT を実装した AI と対戦させた．その結果ある程度のプレイアウトの上では UCT に勝ち越し，可変参照が木探索においても有効に作用することが分かった．

Efficacy on loosely symmetric reasoning with variable reference in Monte Carlo tree search

TOMONOBU NISHIMURA[†1]　KURATOMO OYO[†1]
TATSUJI TAKAHASHI[†2]

This study Was applied to the Monte Carlo tree search to loosely symmetric reasoning proposed with variable reference by Khono. In order to measure the effect of the variable reference, to play against UTC in Reversi. As a result, it was found that on a certain amount of playout is stronger than UCT, the variable reference works effectively even in the tree search.

1. 序論

既存の研究により，ヒトの認知バイアスである対称性バイアスと相互排他性バイアスを緩く含んだ緩対称モデル（*Loosely Symmetric Model*, 以下 LS ） [6]は当たり確率不明の 2 個のスロットマシンを一定回数プレイし報酬の最大化を目指す 2 本腕バンディット問題において，単純な確率を取るモデル等よりも少ない試行回数で正解のマシンを選択出来るようになる事が知られている．LS は本来 2 つの選択肢にしか対応していないが，トーナメント形式を取り，値を比較していく事で一般化する事が可能となる [5]. また LS は確率 0.5 を判断の基準にして選択肢を相対的に評価しているが，甲野は LS を改良しその判断の基準を動的に変化させ, さらに一般化させた LSVR を開発した [4]. LSVR は腕の数が多い n 本腕バンディット問題において，既存の LS や UCB1 [2] といったモデルよりも良い成績を誇る．

本研究では現在，囲碁ゲーム AI 等において有効性が確認されているモンテカルロ木探索に LSVR を実装し, 既存のモデルと戦わせ勝率を調べる事で可変参照型を木探索へ拡張した時の効果を測り，LSVR の実用性を高める事を目的とする．既存のモデルには囲碁 AI において大きな成功を収めた Mogo [3]で採用された UCT [1] を採用した．また可変参照の効果を調べる為に LS をトーナメント方式で一般化し木構造の探索へ拡張させた LSTree [7] を UCT と対戦させ結果を比較した．

†1 東京電機大学大学院
Graduate School of Tokyo Denki University
†2 東京電機大学
Tokyo Denki University

2. n 本腕バンディット問題

n 本腕バンディット問題とは，当たり確率が不明の n 本のスロットマシンを一定回数プレイし，報酬の最大化を目指す意思決定問題の課題である．この問題では当たり確率が高いであろう良いマシンを探索する行動と，現在分かっている最良のマシンをプレイし報酬を得る収穫の行動が考えられる. しかし探索の行動をとれば報酬の収獲は行えず，収獲の行動をとればマシンの探索は行えない．この二つの行動が両立しないトレードオフは探索と収穫のジレンマとして知られている．

n 本腕バンディット問題は様々な問題に当てはめる事が出来，本研究で扱うリバーシでは各盤面の合法手をマシンと考える事で n 本腕バンディット問題に当てはめる事が出来る．

2.1 UCB1

UCB1 は *Auer* らにより発見された n 本腕バンディット問題のマシンの選択アルゴリズムである [2]．このアルゴリズムは全てのマシンを一度プレイし，その後は式 (1) で計算された値が最も高いマシンをプレイしていく. X_i はマシン A_i の期待値，n_i はマシン A_i の選択回数，n は総選

択回数を表している．c は第二項の重みを表し，c が大きいと選択回数の少ないマシンを選択する傾向が強まる．このアルゴリズムは試行回数が十分に多い時正解のマシンを選ぶ事が証明されている．しかしマシンの数が多い時，最初に全てのマシンを一度はプレイしなければならず探索に時間が掛かってしまうという問題もある．

$$UCB1(A_i) = X_i + c\sqrt{\frac{\log n}{n_i}} \quad (1)$$

2.2 LS

篠原らによって提唱された LS について説明する [6]. LS はヒトの認知バイアスである対称性バイアスと相互排他性バイアスを緩く柔軟に調節する因果推論のモデルである．対称性バイアスとは $p \rightarrow q$ から $q \rightarrow p$ を導き，相互排他性とは $p \rightarrow q$ から $\neg p \rightarrow \neg q$ を導くバイアスである．LS は 2 つの選択肢 A, B とその結果として W, ¬W が存在する時，それらの情報を 2×2 分割表に定義しその情報を利用する．例えば a は，選択肢 A を選び結果が W となった回数を表す．

表 1　LS で用いる 2×2 分割表

	結果	
	W	¬W
選択肢 A	a	b
選択肢 B	c	d

この時 LS(W|A) と LS(W|B) は式（2），式（3）で表す事が出来，LS(W|A) と LS(W|B) の値を確率 0.5 を基準として相対的に評価する事により，2 本腕バンディット問題で少ない試行回数ならば UCB1 よりも良い成績を出せることが分かっている．相対評価による推論のため短時間での判断が得意であるが，2 つの選択肢の当たり確率が近い場合の判断は苦手としている

$$LS(W \mid A) = \frac{a + \frac{b}{b+d}d}{a+b+\frac{a}{a+c}c+\frac{b}{b+d}d} \quad (2)$$

$$LS(W \mid B) = \frac{\frac{b}{b+d}d + c}{\frac{c}{a+c}a+\frac{d}{b+d}b+c+d} \quad (3)$$

2.2.1 LST

LS は分割表の情報を使い式（2），式（3）の値を相対的に評価して選択肢を選ぶ．その為，本来は 2 つ以上の選択肢を評価する事は出来ない．しかし選択肢をトーナメント形式で比較する事で n 個の選択肢に一般化することが可能になる [5] ．この時，選択肢が奇数であった場合は最後の選択肢をシード扱いすることで対応する．

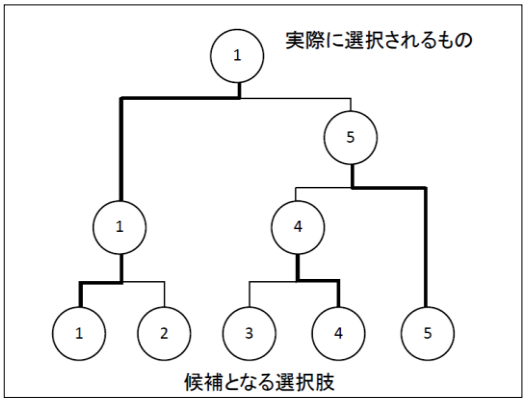

図 1　　LST のトーナメント形式比較

2.2.2 LSVR

本来，LS は確率 0.5 を判断の基準とし選択肢間の良し悪しを比較するが，甲野はこの判断の基準を動的に変化させ複数選択肢へ一般化した LSVR を開発した [4]．LS では表 1 の 2×2 の分割表を利用するが LSVR では以下の表 2 を利用し式（6）で計算される最大の値を持つ選択肢を選ぶ．

表 2　LSVR で用いる分割表

	結果 W	結果¬W
選択肢 A_1	a_1	b_1
選択肢 A_2	a_2	b_2
⋮	⋮	⋮
選択肢 A_{n-1}	a_{n-1}	b_{n-1}
選択肢 A_n	a_n	b_n

$$S_p = \frac{b_{\max} b_{\min}}{b_{\max} + b_{\min}} \quad (4)$$

$$S_n = \frac{a_{\max} a_{\min}}{a_{\max} + a_{\min}} \quad (5)$$

$$LS(W \mid A_i) = \frac{a_i + S_p}{a_i + b_i + \rho_R (S_p + S_n)} \quad (6)$$

$$\rho_R = \frac{1}{R_t} - 1 \quad (7)$$

$$R_{t+1} = \alpha R_t + (1-\alpha) r_t \quad (8)$$

ここで R_t は時刻 t での LSVR の価値基準であるリファ

レンス報酬を表し時刻 t が 0 の時に初期値 0.5 の値をとる．ρ_R は R_t から算出され，実際に LSVR の価値基準を反映させる参照点パラメータである．α はリファレンス報酬に対する学習率で 0.0 から 0.9 の実数とする．α が大きいほど現在の価値基準を重視し，その価値基準からの変化が小さくなる．r_t は時刻 t で得た報酬であり今回扱うリバーシであれば勝ちの時は 1 ，負けの時は 0 を受け取る事になる．LSVR はマシンの数が多い n 本腕バンディット問題で UCB1 や LS よりもかなり良い結果を出している．

3. モンテカルロ木探索

問題に対してランダムな探索を行い，大数の法則からその近似解を得る手法はモンテカルロ法として知られ，その探索に方針を持たせたものはモンテカルロ計画法と呼ばれる．本研究ではモンテカルロ計画法を更に拡張したモンテカルロ木探索を使用する．これは問題を木構造化しその探索を通して問題の最適解を得る手法である．本研究で扱うリバーシであれば，現在の盤面を根ノード，合法手をエッジ，子ノードを各合法手の着手後の盤面とすることでゲーム木として考える事が出来る．この木を深く探索する事はゲームの二手三手先まで予想して打つことに相当する．

3.1 UCT

木構造に対するモンテカルロ法を用いた探索アルゴリズムとしては UCT が有名である[1] . UCT は対象が木構造を持つ時，各ノードから展開される子ノードをスロットマシンと見なした n 本腕バンディット問題であると考え，UCB1 により最良のマシンであると計算された子ノードへ状態を遷移させる．この遷移を根ノードから葉ノードまで行うと最良と考えられる葉ノードを得ることが出来る．葉ノードに到達するとそこで処理を実行し，その結果得た報酬を葉ノードとその全ての親ノードに反映させる．また各葉ノードの処理回数が一定の閾値を超えると木が成長し，そこから新たな子ノードが展開される．今回はリバーシに UCT のアルゴリズムを採用した AI を実装するが，その際には先頭打着緊急度（*first-play urgency*，以下 FPU）の考えを用いる [3]．FPU とは選択肢の中の優先度を表したものであり，一度も試した事の無い選択肢には∞の値が与えられる．一度でも試した選択肢に対しては UCB1 の式により計算された値が与えられる．今回は UCT の中で UCB1 の様に初期状態の入手は行われていないが，FPU の導入により初期状態の入手は自然に達成される事になる．また最終的に選ばれる選択肢は単に試行回数が最も多かった選択肢にしている．UCT は UCB1 と同様にプレイ回数が十分に多い時最良のマシンを選ぶ事が保証されている．また任意のタイミングで探索を終了出来，その性能がそれなりに良い事，木構造の探索において通常のモンテカルロ法をベースにしたアルゴリズムより最良のマシンを得るまでに掛かる時間が短いと行った利点がある．このアルゴリズムはコンピュータ囲碁の AI である MoGo で採用され，大きな成果を出した事で知られている．

3.2 LSTree

因果推論モデルである LS を木構造への対応させた LSTree について説明する [7] . LSTree は UCT と同様の方法により LS を木構造へ拡張した．複数選択肢への一般化には LST を用いており, UCT 同様最終的に選ばれる選択肢は試行回数が一番多いものとする．LSTree は LS の性質を発揮し少ない試行回数では UCT よりよい成績を残している．

4. プレイアウト

モンテカルロ計画法やモンテカルロ木探索の様な統計的手法を用いて盤面競技を AI に競わせる時，プレイアウトという行為が利用される．プレイアウトとは AI がある一手を打った後の展開を，ゲームが終了するまでランダムに進行することである．プレイアウトの結果の勝ち，負けはバンディット問題における報酬 1，0 に相当する．LS であればプレイアウトの結果が勝ちの場合 a に，負けの場合 b に 1 が加算される事となる．UCT ，LSTree といった木構造を組み込んだモデルであれば，末端の葉ノードでプレイアウトが行われる．

5. 報酬の伝播

モンテカルロ木探索では，末端の葉ノードで得た報酬をその親ノードに反映させるが，今回扱くリバーシ等のゲームではその反映を工夫しなければならない．通常の木構造とは違いゲームには自分と相手がおり，木構造上で深くなるたびにそのノードが表す局面の手番は自分，相手と交互に変わっていく事になる．

その為, UCT ならば末端ノードのプレイアウト結果が勝ちであれば，末端ノードの価値に 1 を加算し，その親は逆の手番を意味し負けの結果が返されノードの価値に 0 が加えられる．更にその親ノードは逆の手番を意味するので勝ちの結果の 1 が加算，といった様にプレイアウトの結果の報酬が逆になりながら与えられる. LSTree であればノードを上がっていく毎に 2×2 分割表の a，b に交互に 1 が加算される．

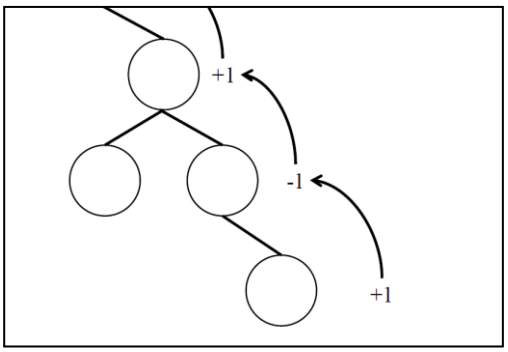

図 2　報酬の伝播の様子

6. 提案手法:LSVR の木探索への実装

本研究では LSVR を木探索へ拡張し，LS の参照点の変更が木探索で有効に作用するかを調べる．本章ではその具体的な実装を述べる．

具体的には LSTree と同様に，UCT の方法を使って木探索へ拡張した．この時，時刻 t でリファレンス報酬 R_t は新たに展開した子ノードに対しその親ノードの値が引き継がれるものとする．これは連続する探索において参照点の学習時間を短縮するためである．

しかし今回は対象がゲーム木の形を取っている為，報酬の伝播同様 LSVR の価値観である参照点も逆転して考えるのが妥当である．その為今回は新たに展開したノードに引き継がれるリファレンス報酬 R_t は $1\text{-}R_t$ とする．

7. シミュレーション

今回は木構造の探索を実装した LSVR の効果を測る為に，同じくモンテカルロ木探索を行う UCT を実装した AI とリバーシで対戦させ勝率を調べた．また動的に参照点を変えることによる効果も調べるため，LSTree と UCT も対戦させた．本章ではシミュレーションの設定及び各モデルのパラメータ等について述べる．

7.1 シミュレーション設定

動的に参照点を変化させる事のない LSTree と UCT，参照点を変える LSVR と UCT を 8×8 のサイズの盤面を持つリバーシで対戦させる．モンテカルロ木探索ではプレイアウトの回数で各モデルの強さが変わってくる為，一手あたりのプレイアウトの回数は 2, 3, 5, 10, 15, 20, 30, 50, 100, 150, 300, 500, 1000, 5000, 10000, 50000 とした．対戦は先手，後手により不利有利が生じる為，それぞれのプレイアウト回数において先手後手入れ替えてそれぞれ 300 局，計 600 回の対局を行いその勝率を調べる．

この時各モデルの木の成長の閾値は 1, 5 回とする．また LSVR の学習率 α は 0.8 ，UCT で用いる UCB1 のパラメータ c の値は 0.1, 0.5 として調べた．

8. シミュレーション結果

シミュレーション結果を以下に示す．縦軸は LSTree, LSVR の UCT に対する勝率を表しており，横軸は一手あたりのプレイアウトの回数を表した片対数グラフとなっている．

ただし図 3, 4 はそれぞれ木の成長の閾値 g を 1, 5 回に，UCB1 のパラメータ c を 0.1 に設定した時の LSTree, LSVR の UCT に対する勝率であり，図 5，6 は閾値 g を 1，5 回に UCB1 のパラメータ c を 0.5 にした時のものである．

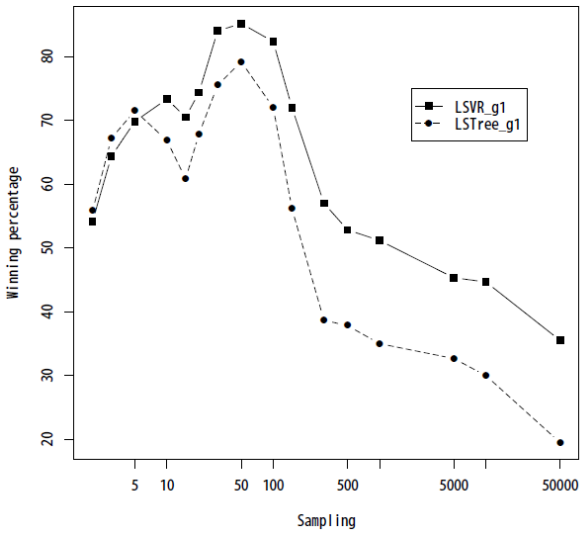

図 3　g = 1，c = 0.1 の時の勝率

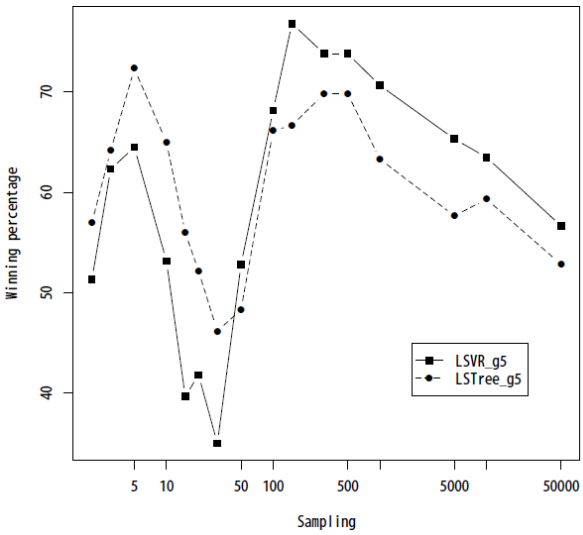

図 4　g = 5，c = 0.1 の時の勝率

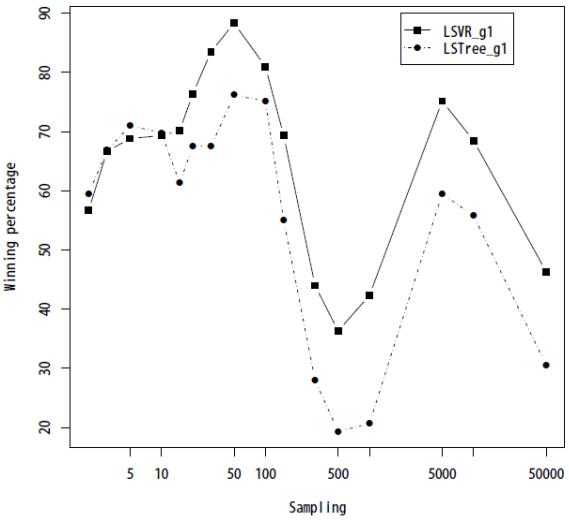

図 5　g = 1，c = 0.5 の時の勝率

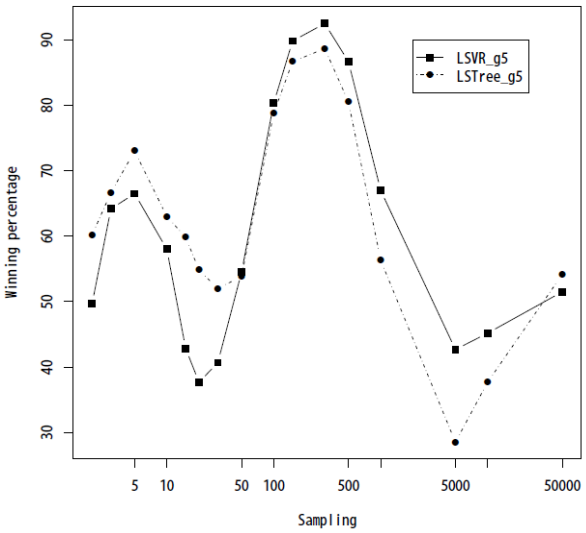

図 6　g = 5，c = 0.5 の時の勝率

9. 考察

シミュレーションの結果, c が 0.5, 木の成長の閾値が 1 の時にはプレイアウトが 300, 500, 1000, 50000 回の時を除き UCT に対して LSVR は勝ち越す事が出来た. そして LSVR は非常に少ないプレイアウト数の時以外では LSTree よりも UCT に対して良い成績を残している. これは動的に参照点を変えた事による効果に他ならない. 300 から 500 回にかけて勝率が大幅に下がってしまう期間については局所解に陥ってしまった様に見えるが，その後勝率が上昇している事から局所解から脱出する事が出来ている．50000 回においても勝率は 50 %を下回るが，これは UCT の試行回数が多い時，正解の選択肢に辿りつく事が出来る性質によるものである．

c が 0.5 ，木の成長の閾値が 5 回の時には，閾値が 1 回の時に比べ，グラフに谷となる形が 2 回現れている．しかしこれはよく見ると閾値が 1 回の時においても非常に軽微であるが現れており，グラフの傾向としては似ている事がわかる. プレイアウトが 1000 回以下の時に LSVR は UCT に対してほぼ勝ち越しており，その後は閾値 1 回の時，局所解に陥りその後脱出している時と同じ様な形となった．

次に c が 0.1 の時であるが，成長の閾値が 5 の時はやはり最初に勝率のグラフに谷が現れるが 50000 回のプレイアウト時でも LSVR は勝ち越しており非常に良い結果を残せたと言える．

c が 0.1，成長の閾値が 1 回の時には設定が最も UCT にフィットした時である事がわかり，多くのプレイアウト数において LSVR や LSTree に勝ち越している．しかしこの様な UCT にフィットした場合にあっても, 少ないプレイアウト数では勝率は 8 割を超す事が出来た．

4 つの図を比べると，やはり全体的にプレイアウト数が少ない時には LSTree や LSVR が強く，プレイアウト数が多い時には UCT が強い事が確認出来る. また LSTree よりも LSVR が多くの場合において良い成績を出しており, 木探索においても動的な参照点の変更に効果がある事がわかる．

なお全てのケースで極少ないプレイアウト数で LSTree が LSVR を上回っている．この現象は単純なバンディット問題のシミュレーションでも確認されており参照点の学習によるものと考えられている．

10. 展望

今回の研究結果により木構造の探索においても LSVR の動的な参照点の変更が効果的に作用する事を確認できた. 今回の研究ではリバーシについて扱ったが，LSVR は多くの選択肢で良い結果を出す事が分かっており合法手の少ないリバーシは得意とは言えない. 一方 UCB1 は多くの選択肢を苦手としている．よって今後合法手の多い囲碁のような競技に実装することで，LSVR の多くの選択肢においても高い正解率を出せるという長所を生かした成果を期待できる. また今回のリバーシでは 8×8 の盤面でシミュレーションを行ったが, これが 10×10 等の広さを持ち木が深くなり探索空間が広がった場合にも，LSVR の参照点の変更が有効に作用する可能性も考えられる．

他に UCT は合法手の多い囲碁では，選択肢を減らす為に囲碁の知識を利用し不必要ものを削っているが，LSVR の多くの選択肢に強いという性質を利用し選択肢を絞り，問題に対する特定知識を利用しない探索のメタ戦略としての成果も期待できる．また探索時間が短いステップの間は LSVR を用い，探索ステップのある程度増大した時 UCT を使い両者の長所を生かすといったハイブリッドなモデルが考えられる．

謝辞　本研究にご協力頂いた皆様に，謹んで感謝の意を表します．ありがとうございました．

参考文献

1) Levente Kocsis, Csaba Szepesvari: Bandit based Monte-Carlo Planning, ECML'06 In: ECML-06, *LNCS*, 4212, pp. 282-293 (2006).
2) Peter Auer, Nicolo Casa-Bianchi, and Paul Fischer: Finite-time analysis of the multiarmed bandit problem, *Machine Learning*, 47, 235-256 (2002).
3) Sylvain Gelly, Yizao Wang, Remi Munos, Olover Teytaud,: Modification of UCT with Patterns in Monte-Carlo Go, *INRIA*, 6062 (2006).
4) Yu Kohno, Tatsuji Takahashi: Loosely Symmetric Reasoning to Cope with The Speed-Accuracy Trade-off, *The 6th International Conference on Soft Computing and Intelligent Systems, The 13th International Symposium on Advanced Intelligent Systems*, 投稿中.
5) 大用 庫智: ヒト認知バイアスのモンテカルロ法への応用, 2009 年度情報科学科卒業文集 (2010).
6) 篠原 修二, 田口 亮, 桂田 浩一, 新田 恒雄: 因果性に基づく信念形成モデルと N 本腕バンディット問題への適用, 人工知能学会論文誌, 22 巻 1 号, pp.58-68 (2007).

7) 西村 友伸: 緩い対称性推論のモンテカルロ木探索での効果, 2011 年度情報システムデザイン学系卒業文集 (2012).

著者紹介

西村友伸

東京電機大学大学院

1989 年東京生まれ. 東京電機大学理工学科情報システムデザイン学系を卒業し同大学大学院理工学研究科情報学専攻に在籍. ヒトの認知モデルを用いた短時間での木構造の探索を研究中.

大用庫智

東京電機大学大学院

1987 年東京生まれ. 東京電機大学理工学部情報科学科, 東京電機大学大学院理工学研究科情報学専攻を経て現在東京電機大学大学院先端科学研究科情報学専攻に在学中. 人工知能学会, 認知科学会会員.

高橋達二

東京電機大学

1978 年生まれ. 東京電機大学理工学部助教, 東北大学共同研究員. 人間の推論に遍在する対称性を軸とし, 認知の柔軟さと創造性を, 自己言及やフレーム問題の関係において可能な限り経験的に研究. 人工知能学会, 認知科学会など会員.