

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 7 月 31 日(31.07.2014)



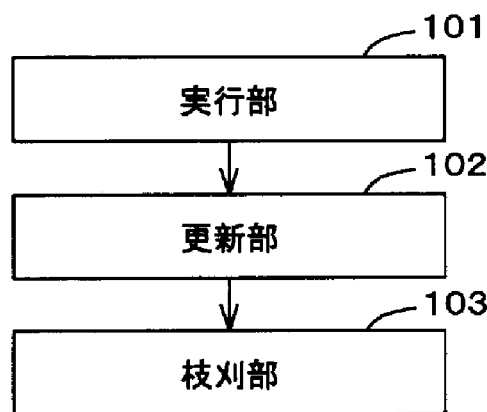
(10) 国際公開番号
WO 2014/115232 A1

- (51) 国際特許分類:
G06N 5/04 (2006.01) *G06F 19/00* (2011.01)
- (21) 国際出願番号: PCT/JP2013/007673
- (22) 国際出願日: 2013 年 12 月 27 日(27.12.2013)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2013-011628 2013 年 1 月 25 日(25.01.2013) JP
- (71) 出願人: 日本電気株式会社(NEC CORPORATION) [JP/JP]; 〒1088001 東京都港区芝五丁目 7 番 1 号 Tokyo (JP).
- (72) 発明者: 白木 孝(SHIRAKI, Takashi); 〒1088001 東京都港区芝五丁目 7 番 1 号日本電気株式会社内 Tokyo (JP).
- (74) 代理人: 岩壁 冬樹, 外(IWAKABE, Fuyuki et al.); 〒1040031 東京都中央区京橋二丁目 8 番 7 号 読売中公ビル 6 階 サンライズ国際特許事務所 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: SOLUTION SEARCH DEVICE, SOLUTION SEARCH METHOD, AND SOLUTION SEARCH PROGRAM

(54) 発明の名称: 解探索装置、解探索方法および解探索プログラム



101 Execution unit
102 Updating unit
103 Pruning unit

(57) Abstract: Provided are a solution search device, a solution search method, and a solution search program which are capable of reliably calculating a solution within a specified calculation time and improving the accuracy of the solution without causing the amount of memory usage to increase when solving an optimization problem, etc. The present invention includes: an execution unit (101) for selecting a node which is an object for simulation to be executed from among nodes constituting choices in a search tree of a simulation-based solution search, and executing a simulation from the selected node; an updating unit (102) for calculating an evaluation value using an evaluation function on the basis of the simulation result, and updating the evaluation value of the selected node and that of a host node thereof; and a pruning unit (103) for cutting off from the search tree a node having an evaluation value which does not satisfy a predetermined criterion.

(57) 要約:

[続葉有]



添付公開書類:

— 国際調査報告 (条約第 21 条(3))

最適化問題などを解く際に、メモリ使用量を増大させることなく、指定された計算時間内に確実に解を算出することができ、かつ解の精度を向上させることができる解探索装置、解探索方法および解探索プログラムを提供する。シミュレーションを用いた解探索において、探索木中の選択肢となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行する実行部 101 と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新する更新部 102 と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す枝刈部 103 とを含む。

明 細 書

発明の名称： 解探索装置、解探索方法および解探索プログラム 技術分野

[0001] 本発明は、最適化計算などにおける解探索に適用される解探索装置、解探索方法および解探索プログラムに関する。

背景技術

[0002] 最適化問題は、目的関数と制約条件が設定され、目的関数が最善となる最適解一つを導出する問題であることが多い。人工知能の分野では、非特許文献1に記載されたMCTS（モンテカルロ木探索（Monte-Carlo Tree Search））など、シミュレーションを用いた探索方法が注目されている。それらの探索方法は、データマイニングや機械学習の分野で注目されているMBP（Multi-armed Bandit Problem）を解く方法の発展形として位置付けられる。また、それらの探索方法は、実用化されつつあり、既に実用化に成功した例として、コンピュータ囲碁がある。OR（Operations Research）などで用いられる最適化においては、それらの探索方法の実用化が期待されているが、実用化が難しい。

[0003] コンピュータ囲碁と最適化との最も異なる点は、解空間の木（以下、解空間木という。）において、コンピュータ囲碁などは解空間木の各段において次の段の中で最も良いノードを探すことを目的とするのに対して、最適化では最下段の解ノードの中で最も良いノードを探すことを目的とすることにある。解空間木の最下段まで探索木を伸ばすことは、MCTSにおいて今までにない目的となる。

[0004] MCTSで重要となる技術は、シミュレーション結果を用いた各ノードの評価関数の設定方法と、プレイアウト（Playout）と呼ばれる1回のシミュレーションをする方法である。コンピュータ囲碁で用いられる評価関数は、非特許文献2に記載されたUCB（Upper Confidence

e Bound) を元にしたものである。コンピュータ囲碁で用いられる評価関数では、期待値をベースにしたものが用いられる。期待値は、囲碁における勝率などである。しかし、期待値よりも、最適な解や、上位 k 個の解、つまり k 番目までに良い解の平均を評価関数に用いる方が良いとされている（非特許文献 3 参照。）。つまり、コンピュータ囲碁では、これらの評価関数を適切に選択して用いることが重要となる。以下、期待値にもとづく評価関数を `mean`、最適な解にもとづく評価関数を `best`、上位 k 個の解の平均にもとづく評価関数を `kbest` と表現する。

[0005] `best` や `kbest` のどちらを用いる方が良いかは最適化を行う対象によって異なるため、一意には定めることができない。つまり、どの評価関数を用いるかを適切に選択することが重要となる。

先行技術文献

非特許文献

[0006] 非特許文献 1: C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Travençolo, D. Perez, S. Samothracis and S. Colton, A Survey of Monte Carlo Tree Search Methods, IEEE Transactions on Computational Intelligence and AI in Games, Vol. 4, No. 1, March 2012.

非特許文献 2: P. Auer, N. Cesa-Bianchi, and P. Fischer, Finite-time Analysis of the Multiarmed Bandit Problem, Machine Learning, Vol. 47, p. 235–256, 2002

非特許文献 3: A. Rimmel, F. Teytaud, and T. Cazenave, "Optimization of the n

ested Monte-Carlo algorithm on the traveling salesman problem with time windows," in Proc. Appl. Evol. Comput. 2, Torino, Italy, 2011, pp. 501-510

発明の概要

発明が解決しようとする課題

- [0007] 解空間木が大きい時に、計算初期から探索木を拡大すると保有するノード数が増大する。その際、拡大する基準を緩くすると、メモリ容量がオーバーする可能性がある。また、拡大する基準を厳しくすると、膨大な計算時間を費やしても解空間木の底にある解ノードに達しない可能性がある。
- [0008] 最適化計算では、一般的に、解を取得したい時間を予めユーザ等が指定する。MCTSを最適化で成功させることを目標とすると、探索木を解空間木の底に指定時間までに到達させなければ、解の精度は上がらない。つまり、解の精度を向上させるには、指定時間までに確実に探索木を解空間木の底に到達させる必要がある。
- [0009] そこで、本発明は、最適化問題などを解く際に、メモリ使用量を増大させることなく、指定された計算時間内に確実に解を算出することができ、かつ解の精度を向上させることができる解探索装置、解探索方法および解探索プログラムを提供することを目的とする。

課題を解決するための手段

- [0010] 本発明による解探索装置は、シミュレーションを用いた解探索において、探索木中の選択枝となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行する実行部と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新する更新部と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す枝刈部とを含むことを特徴とする。

[0011] 本発明による解探索方法は、シミュレーションを用いた解探索において、探索木中の選択枝となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行し、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新し、予め定められた基準を満たさない評価値を持つノードを探索木から切り離すことを特徴とする。

[0012] 本発明による解探索プログラムは、コンピュータに、シミュレーションを用いた解探索において、探索木中の選択枝となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行する処理と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新する処理と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す処理とを実行させることを特徴とする。

発明の効果

[0013] 本発明によれば、最適化問題などを解く際に、メモリ使用量を増大させることなく、指定された計算時間内に確実に解を算出することができ、かつ解の精度を向上させることができる。

図面の簡単な説明

[0014] [図1]最適化システムの第1の実施形態の構成を示すブロック図である。

[図2]探索木に対して枝刈が実行される様子を示す説明図である。

[図3]第1の実施形態における計算部の動作を示すフローチャートである。

[図4]最適化システムの第1の実施形態の他の構成を示すブロック図である。

[図5]枝刈部を2つ含む最適化装置を備えた最適化システムの構成の一例を示すブロック図である。

[図6]図5に示す最適化装置における計算部の動作を示すフローチャートである。

[図7]本発明による解探索装置の最小構成を示すブロック図である。

発明を実施するための形態

[0015] 実施形態 1.

以下、本発明の第 1 の実施形態を図面を参照して説明する。

[0016] 図 1 は、最適化システムの第 1 の実施形態の構成を示すブロック図である。

[0017] 図 1 に示すように、第 1 の実施形態における最適化システムは、ユーザ端末 1 と、最適化装置 2 とを備える。ユーザ端末 1 と最適化装置 2 とは、通信可能に接続される。なお、図 1 には 1 つのユーザ端末が例示されているが、ユーザ端末は最適化装置 2 にいくつ接続されていてもよい。

[0018] ユーザ端末 1 は、例えばパーソナルコンピュータ等の情報処理端末である。ユーザ端末 1 は、操作部 1 1 と、表示部 1 2 とを含む。

[0019] 操作部 1 1 は、実行する最適化計算に必要な情報（以下、最適化計算入力情報という。）を入力する。また、操作部 1 1 は、実行指示を入力する。操作部 1 1 は、最適化計算入力情報とともに実行指示を最適化装置 2 に出力する。

[0020] 表示部 1 2 は、最適化装置 2 から最適化計算結果の解を受け取り、表示する。

[0021] 最適化装置 2 は、GUI (Graphical User Interface) 部 2 1 と、計算部 2 2 と、記憶部 2 3 とを含む。

[0022] GUI 部 2 1 は、ユーザ端末 1 の操作部 1 1 から、最適化計算入力情報を受け取る。GUI 部 2 1 は、最適化計算入力情報を計算部 2 2 に伝達する。GUI 部 2 1 は、計算部 2 2 から最適化計算結果の解を受け取り、ユーザ端末 1 の表示部 1 2 へ伝達する。

[0023] 計算部 2 2 は、選択部 2 2 1 と、拡大部 2 2 2 と、シミュレーション部 2 2 3 と、評価値更新部 2 2 4、枝刈部 2 2 5 とを含む。

[0024] 選択部 2 2 1 は、展開されたノードの中からプレイアウトの実行対象となるノードを選択する。以下、プレイアウトの実行対象となるノードを選択ノードという。

- [0025] 拡大部 2 2 2 は、探索木（ツリー）を拡大する。具体的には、拡大部 2 2 2 は、予め定められた基準に従って、選択部 2 2 1 で選ばれたノードを展開する必要があるか否かを判断し、必要となればさらに一段下位にノードを展開して、探索木を拡大する。拡大部 2 2 2 は、ノードを展開した場合には、選択ノードを当該一段下位のノードに選択し直す。
- [0026] シミュレーション部 2 2 3 は、シミュレーションを実行する。具体的には、シミュレーション部 2 2 3 は、プレイアウト、すなわち、ランダムシミュレーションのような単純な方法で 1 つの解を探索し、解の評価値を取得する。
- [0027] 評価値更新部 2 2 4 は、シミュレーション部 2 2 3 が行ったプレイアウトの結果により、各ノードの解の評価値を更新する。各ノードの評価値は、繰り返し実行されるシミュレーションで得られた評価値を集めた統計値からなり、評価値更新部 2 2 4 は当該統計値を更新する。
- [0028] 評価値更新部 2 2 4 は、指標値を算出する。本実施形態では、`best`、`mean`、`k best` の評価関数値（以下、解の評価値ともいう。）を指標値として、それぞれ算出する。
- [0029] 評価値更新部 2 2 4 は、算出された評価値を組み合わせる総合評価値を算出する。その際、評価値更新部 2 2 4 は、各指標（`best`、`mean`、`k best`）のトレードオフを考慮し、また各指標で特長を持つ値が強調されるように、総合評価値の算出を行う。
- [0030] 評価値更新部 2 2 4 は、総合評価値の算出結果をもとに、各ノードの評価値を更新する。
- [0031] 枝刈部 2 2 5 は、枝刈（`Pruning`）を実行する。図 2 は、探索木に対して枝刈が実行される様子を示す説明図である。図 2 に示す三角形の枠は解空間木を表す。三角形の底辺を囲う楕円は、解空間木の最下段のノード、つまり解ノードを表す。網点の領域は、探索木を表す。黒丸は、探索木の根ノード、すなわち解空間木の最上段のノードを表す。白丸は、探索木のリーフノードを表す。図 2 に示すように、本実施形態では、時間の経過とともに

、また、探索木の展開とともに、枝刈が解空間木の最下段に向かって実行される。また、枝刈部 225 は、探索木の大きさ（図 2 に示す網点の領域の面積）が一定の大きさを超えないように、枝刈を実行する。

[0032] 記憶部 23 は、目的関数や制約条件を記憶する。最適化システムがスケジューリング問題に適用される場合には、記憶部 23 は、タスク情報や担当情報など、問題を解くために必要なデータ（以下、問題データという。）を記憶する。また、記憶部 23 は、計算部 22 での計算処理が進む際に、ノードの評価値など、変化する情報を記憶する。本実施形態では、記憶部 23 は、計算部 22 が各計算途中で得たノードの探索回数や評価値を記憶する。また、記憶部 23 は、計算部 22 で求められた解の中で保持する必要がある解を記憶する。

[0033] なお、GUI 部 21、計算部 22 は、例えば、解探索プログラムに従って動作するコンピュータによって実現される。この場合、最適化装置 2 が備える CPU が解探索プログラムを読み込み、そのプログラムに従って、GUI 部 21 および計算部 22 として動作する。また、GUI 部 21 および計算部 22 の各部は別々のハードウェアで実現されていてもよい。

[0034] また、記憶部 23 は、最適化装置 2 が備える RAM (Random Access Memory) 等の記憶装置によって実現される。

[0035] 次に、本実施形態の動作を説明する。

[0036] 図 3 は、第 1 の実施形態における計算部 22 の動作を示すフローチャートである。

[0037] ここでは、図 1 に示す最適化システムがスケジューリング問題に適用される場合を例にする。

[0038] まず、ユーザがユーザ端末 1 の操作部 11 に対して、最適化計算入力情報を入力する。ユーザは、最適化計算をしたいタスク、従事可能な担当者、各担当者がそれぞれのタスクに従事した時のコストや有効性などの問題データを最適化計算入力情報として入力する。このとき、ユーザは、最適化計算入力情報とともに実行指示を操作部 11 に入力する。操作部 11 は、最適化計

算入力情報と実行指示とを最適化装置 2 に出力する。

- [0039] 最適化装置 2 の GUI 部 2 1 は、ユーザ端末 1 から最適化計算入力情報とともに実行指示を受け取ると、最適化計算入力情報を計算部 2 2 へ伝達する。計算部 2 2 は、前処理として最適化計算入力情報を入力する（ステップ S 3 0 1）。
- [0040] ステップ S 3 0 1 の後、計算部 2 2 の選択部 2 2 1 は、展開されたノードの中から、シミュレーションすべきノードを選択する（ステップ S 3 0 2）。なお、初期状態ではノードは 1 つのみであるので、そのノードが選択対象となる。ノードの選択は、本実施形態では、評価値更新部 2 2 4 によって算出された指標値を用いて行う。
- [0041] 拡大部 2 2 2 は、選択部 2 2 1 で選択されたノードのプレイアウト回数が、事前に定められた条件を満たした時に、探索木を 1 段下位のノードまで展開する（ステップ S 3 0 3）。本実施形態では、拡大部 2 2 2 は、当該プレイアウト回数が予め定められた回数を超えた時に展開する。なお、初期状態でノードが 1 つのみである時は、この条件に関わらず展開する。展開した場合には、拡大部 2 2 2 は、展開したノードのうちの 1 つを選択ノードとする。
- [0042] シミュレーション部 2 2 3 は、選択ノードからプレイアウト、つまりランダムシミュレーションを実行し解を 1 つ探索する（ステップ S 3 0 4）。なお、1 つの選択ノードに対して複数のシミュレーションを実行し複数の解を探索することも可能であるが、ここでは、もっともシンプルな例として 1 つの選択ノードに対して 1 つのシミュレーションを実行し、1 つの解を探索する方法について説明する。本発明の技術的範囲は、1 つの選択ノードに対して 1 つのシミュレーションを実行する形態に限定されない。従って、1 つの選択ノードに対して複数のシミュレーションを実行する形態も本発明の技術的範囲に含まれ得る。
- [0043] 評価値更新部 2 2 4 は、シミュレーション部 2 2 3 によって得られた解を用いて、best, mean, kbest などの解の評価値を更新する。な

お、本実施形態では、評価値更新部 224 は、mean の解の評価値を更新する（ステップ S305）。best の解の評価値は、現在までにその選択ノードから実行されたシミュレーションの結果の中で最適なものである。mean の解の評価値は、現在までにその選択ノードから実行されたシミュレーションの結果を平均したものである。k best の解の評価値は、現在までにその選択ノードから実行されたシミュレーションの結果の中で k 番目までに良い解を平均したものである。

[0044] 評価値更新部 224 は、指標値である、mean の解の評価値について、以下の正規化の処理を行う。

[0045] 選択ノードとその兄弟ノード（選択ノードと共通の親ノードをもつノード）における指標値をそれぞれ $v_1, v_2, v_3, \dots, v_L$ とした時、最大値 M 、最小値 m は以下のように表すことが可能である。なお、 L は、共通の親ノードをもつ子ノードの総数である。

[0046] [数1]

$$M = \max_{1 \leq i \leq L} v_i \quad \dots (式1)$$

[0047] [数2]

$$m = \min_{1 \leq i \leq L} v_i \quad \dots (式2)$$

[0048] ここで、 v_i は、予め定められた基準により、最小値 m に近い場合に良い値であると判断されたとする。また、最大値 M に近い場合に悪い値であると判断されたとする。評価値更新部 224 は、良い値である最小値 m を 0、悪い値である最大値 M を 1 とするために、 v_i を正規化する。具体的には、評価値更新部 224 は、 v_i を次のように変換する。Value_i は、 v_i の正規化後の値である。

[0049] [数3]

$$Value_i = \begin{cases} \frac{v_i - m}{M - m}, & \text{if } M > m \\ 1, & \text{if } M = m \end{cases} \quad \dots (式3)$$

- [0050] なお、 $Value_i$ の値が一定の分散におさまるように、 v_i をさらに正規化してもよい。
- [0051] 評価値更新部224が、 $mean$ について上記の処理を実行することにより、 $mean$ に対応する $Value_i$ が算出される。以下、 $best$ 、 $mean$ 、 $kbest$ に対応する $Value_i$ を、それぞれ $bestValue$ 、 $meanValue$ 、 $kbestValue(k)$ と表現する。なお、“ (k) ”は、上位 k 個の解を平均算出の対象とすることを表す。
- [0052] 評価値更新部224は、 $meanValue$ を、選択ノードとその上位のノードそれぞれで計算し、算出結果をもとにそれぞれのノードの評価値を更新する。
- [0053] 枝刈部225は、探索木の大きさ、例えば、探索木全体のノード数や探索木のリーフノードの数が大きく、枝刈が必要であると判断した時に、枝刈を実行する。また、与えられた指定時間内に探索木を解空間木の底に到達させるために、探索する空間を狭める必要があると判断した時に、枝刈を実行する（ステップS306）。本実施形態では、枝刈部225は、ステップS305において評価値更新部224が更新した $mean$ で各ノードを評価し、評価値の悪いノードは探索木から枝刈する。その後、そのノードやその下位のノードからシミュレーションされることはない。
- [0054] ステップS306における、指定時間内に探索木を解空間木の底に到達させるための枝刈の実行方法として、各段に対する枝刈の実行間隔を調整する方法がある。解空間木全体における深さのレベルが、1段目から底（ N 段目）までであるとしたとき、計算時間が経過するとともに探索木の中心は底に向かって移動する。そこで、枝刈部225は、計算時間全体の n/N が経過した時に n 段目の深さに位置するノードの数を抑える枝刈を実行する。例えば、 $N=10$ である場合、許された計算時間の60%が経過した時には6段目の深さに位置するノードの数を抑える枝刈を実行する。これにより、計算時間内に探索木を解空間木の底に到達させることができる。なお、各段に対する枝刈の実行間隔は均等でなくてもよい。例えば、計算時間全体の $\sqrt{n/N}$

）が経過した時に n 段目の深さに位置するノードの数を抑える枝刈をしてもよい。それにより、計算時間をより多く必要とする根ノード付近のノードに対して、より多くの計算時間を割り当てることができる。

[0055] 計算部 22 は、計算部 22 における計算時間が事前に定められた上限に達するまで、ステップ S302～S306 の処理（選択処理、ノード展開処理、シミュレーション実行処理、評価値更新処理および枝刈処理）を繰り返し実行する。つまり、当該計算時間が上限に達していない場合は（ステップ S307 の Yes）、計算部 22 はステップ S302 の処理に戻る。当該計算時間が上限に達した場合は（ステップ S307 の No）、計算部 22 は、計算を終了し、最適化計算結果、つまり探索して得た解を示す解情報を GUI 部 21 に渡す（ステップ S308）。なお、計算部 22 は、計算時間ではなく、要件として与えられた解の値が算出されるまでステップ S302～S306 の処理を繰り返し実行するようにしてもよい。

[0056] また、ステップ S302～S306 の計算処理において、計算部 22 は、各計算途中で得たノードの探索回数や評価値を含む情報を記憶部 23 に格納する。また、計算部 22 は、探索して得た解を含む情報を記憶部 23 に格納する。計算部 22 は、記憶部 23 に格納された情報を取得することにより、計算途中における各ノードの探索回数や評価値を認識することができる。

[0057] なお、本実施形態では、問題データがユーザ端末 1 から最適化計算入力情報として計算部 22 に入力される場合を例にしたが、計算部 22 は記憶部 23 に格納された問題データを取得するようにしてもよい。そのような形態を実現するには、ユーザ等が予め問題データを記憶部 23 に格納すればよい。

[0058] また、ステップ S306 の枝刈処理を実行するタイミングは、ステップ S305 の評価値更新処理の後に限られない。例えば、図 4 に示すように、拡大部 222 とシミュレーション部 223 との間に枝刈部 225 を配置するようにしてもよい。つまり、ステップ S303 の処理とステップ S304 の処理との間に、ステップ S306 の枝刈処理を実行するようにしてもよい。図 4 は、最適化システムの第 1 の実施形態の他の構成を示すブロック図である

。

[0059] 以上に説明したように、本実施形態によれば、MCTSにおいてシミュレーション結果を用いた枝刈をすることにより、適切に探索範囲を狭めることができる。それにより、メモリ使用量を増大させることなく、与えられた計算時間内で探索木を解空間木の底まで到達させることができ、確実に解を算出することができる。また、シミュレーション結果をもとに重要であると判断したノードのシミュレーション回数を増やすことができる。よって、MCTSなどのシミュレーションによって評価値を算出する探索方法の解の精度を改善する可能性を高めることができる。また、解の精度を改善することで、実用化が難しいORなどについて、実用化の可能性を高めることができる。

[0060] 実施形態2.

以下、本発明の第2の実施形態を説明する。

[0061] 第2の実施形態における最適化システムの構成は、第1の実施形態の構成と同様である。

[0062] ここでは、第1の実施形態と同様に、最適化システムがスケジューリング問題に適用される場合を例にする。

[0063] 第2の実施形態における計算部22の動作は、図3に示す第1の実施形態の動作と同様である。

[0064] しかし、ステップS302における選択部221の動作、ステップS305における評価値更新部224の動作およびステップS306における枝刈部225の動作が異なる。ここで、ステップS302、ステップS305およびステップS306における動作を説明する。

[0065] ステップS302において、選択部221は、ノードを選択する際に、枝刈部225が枝刈を実行するか否かを判断するために用いる指標値とは異なる指標値を用いる。本実施形態では、ノードの選択に用いる指標をmean、枝刈に用いる指標をbestとする。すなわち、ノードの選択に用いる指標値をmeanValue、枝刈に用いる指標値をbestValueとす

る。

[0066] ステップS305において、評価値更新部224は、式1～式3を用いて、`meanValue`および`bestValue`を算出する。このとき、評価値更新部224は、`meanValue`および`bestValue`を、選択ノードとその上位のノードそれぞれで計算し、算出結果をもとにそれぞれのノードの評価値を更新する。

[0067] ステップS306において、枝刈部225は、選択部221がノード選択に用いる指標とは異なる指標`best`で各ノードを評価し、枝刈処理を実行する。

[0068] このように、本実施形態では、探索木内の各ノードは、ノード選択のための評価値である`meanValue`と、枝刈のための評価値である`bestValue`との両方を持つ。具体的には、評価値更新部224が、探索木内の各ノードに対応づけて、`meanValue`および`bestValue`を記憶部23に格納する。

[0069] なお、ノード選択のための評価値である`meanValue`を、そのノードとその親とを結ぶエッジ（枝）に持たせ、枝刈のための評価値である`bestValue`をノードに持たせるようにしてもよい。

[0070] 以上に説明したように、本実施形態では、第1の実施形態と同様の効果を得つつ、枝刈に用いる指標に、ノード選択とは別の評価関数を設定することができる。それにより、枝刈において、価値の低いノードをより適切に選択し、取り除くことができる。例えば、枝刈において、それまでに最適な結果（`bestValue`）を出したノードは、平均値（`meanValue`）が低くても取り除くことが適さないと考えられる。本実施形態によれば、そのようなノードが取り除かれることを回避することができる。

[0071] このように、より適切に価値の低いノードを取り除くことにより、実行可能なメモリ空間の中で、より価値の高いノードのシミュレーション回数を増やすことができる。より重要なノードのシミュレーション回数をより適切に増やすことができれば、MCTSなどのシミュレーションによって評価値を

算出する探索方法の解の精度をさらに改善する可能性を高めることができる。

[0072] 実施形態 3.

以下、本発明の第 3 の実施形態を説明する。

[0073] 第 3 の実施形態における最適化システムの構成は、第 1 の実施形態の構成と同様である。

[0074] しかし、最適化装置 2 の評価値更新部 224 は、各指標値 $bestValue$ 、 $meanValue$ 、 $kbestValue$ をもとに、ノードの総合評価値を算出する。そして、最適化装置 2 の評価値更新部 224 は、算出したノードの総合評価値により、各ノードの評価値を更新する。

[0075] ここでは、第 1 の実施形態と同様に、最適化システムがスケジューリング問題に適用される場合を例にする。

[0076] 第 3 の実施形態における計算部 22 の動作は、図 3 に示す第 1 の実施形態の動作と同様である。

[0077] しかし、ステップ S305 における評価値更新部 224 の動作が異なる。ここで、ステップ S305 における評価値更新部 224 の動作を説明する。

[0078] 評価値更新部 224 は、ステップ S304 の後、式 1～式 3 を用いて、 $meanValue$ 、 $bestValue$ 、 $kbestValue(k)$ を算出する。

[0079] 評価値更新部 224 は、ノードごとに次の計算式を用いてノードの総合評価値を得る。 $hValue$ は、ノードの総合評価値である。 w 、 ε は、各指標 $best$ 、 $mean$ 、 $kbest$ の重みを表す係数である。

[0080] [数 4]

$$hValue(bestValue, meanValue, kbestValue(k)) = \frac{w_{best} \left(\frac{1}{1 + \varepsilon_{best}} \right) + w_{kbest(k)} \left(\frac{1}{1 + \varepsilon_{kbest(k)}} \right) + w_{mean} \left(\frac{1}{1 + \varepsilon_{mean}} \right)}{w_{best} \left(\frac{1}{bestValue + \varepsilon_{best}} \right) + w_{kbest(k)} \left(\frac{1}{kbestValue(k) + \varepsilon_{kbest(k)}} \right) + w_{mean} \left(\frac{1}{meanValue + \varepsilon_{mean}} \right)}$$

[0081] 例えば、 w_{best} 、 w_{mean} 、 $w_{kbest(k)}$ を大きくすると、式全体に対する

各指標の影響度、つまり $hValue$ に対する各指標の影響度を強くすることができる。

[0082] ε_{best} , ε_{mean} , $\varepsilon_{kbest(k)}$ は、 $bestValue$, $meanValue$, $kbestValue(k)$ がそれぞれ最も良い時、つまりゼロ付近にある時の各指標の影響度を調整する係数である。各指標の影響度を強くしたい場合には、 ε_{best} , ε_{mean} , $\varepsilon_{kbest(k)}$ の値を小さくする、つまりゼロに近づける。例えば、 $bestValue$ がゼロ付近にある場合、 $\varepsilon_{best} = 0.01$ であれば、分母の $best$ の項は「 $w_{best} \times 100$ 」になる。 $\varepsilon_{best} = 0.1$ であれば、分母の $best$ の項は「 $w_{best} \times 10$ 」になる。このように、 $bestValue$ がゼロ付近にある時に、 ε_{best} を小さくすればするほど、指標 $best$ の影響度が強くなる。 ε を調整することにより、 $bestValue$, $meanValue$, $kbestValue(k)$ が良い値であるときに、総合評価値 $hValue$ を、最も良い値であるゼロに到達させる速さを調節することが可能となる。

[0083] このように、 w 、 ε の値を用いることにより、各指標のトレードオフを考慮できる。また、特定の指標に強みを持つノードの評価値をより良くすることができる。なお、 w 、 ε は、予め最適化装置 2 の記憶部 23 に格納される。

[0084] また、評価値更新部 224 は、 w や ε の値を、ノード選択に用いる評価関数と、枝刈に用いる評価関数とで区別する。例えば、ノード選択に用いる評価関数では、

[0085] [数5]

$$w_{best} = 0.1, \varepsilon_{best} = 0.1$$

$$w_{kbest(k)} = 0.2, \varepsilon_{kbest(k)} = 0.04$$

$$w_{mean} = 0.7, \varepsilon_{mean} = 0.01$$

[0086] とする。枝刈に用いる評価関数では、

[0087]

[数6]

$$w_{best} = 0.5, \varepsilon_{best} = 0.001$$

$$w_{kbest(k)} = 0.3, \varepsilon_{kbest(k)} = 0.01$$

$$w_{mean} = 0.2, \varepsilon_{mean} = 0.1$$

[0088] とする。

[0089] 評価値更新部 224 は、ノード選択に用いる評価関数値と枝刈に用いる評価関数値を、選択ノードとその上位のノードそれぞれで計算し、算出結果をもとにそれぞれのノードの評価値を更新する。

[0090] 以上に説明したように、本実施形態では、第 2 の実施形態と同様の効果を得ることができる。

[0091] 実施形態 4.

以下、本発明の第 4 の実施形態を説明する。

[0092] 第 4 の実施形態における最適化システムの構成は、第 1 の実施形態の構成と同様である。

[0093] しかし、最適化装置 2 の評価値更新部 224 は、第 3 の実施形態と同様に、各指標値 $bestValue$ 、 $meanValue$ 、 $kbestValue$ をもとに、ノードの総合評価値を算出する。そのとき、評価値更新部 224 は、探索木の深さのレベルごとに、ノードの総合評価値を算出する。なお、解空間木は N 段のレベルを有するとする。つまり、解空間木は N 段の深さを有するとする。

[0094] ここでは、第 1 の実施形態と同様に、最適化システムがスケジューリング問題に適用される場合を例にする。

[0095] 第 4 の実施形態における計算部 22 の動作は、第 3 の実施形態の動作と同様である。

[0096] しかし、ステップ S305 における評価値更新部 224 の動作が異なる。ここで、ステップ S305 における評価値更新部 224 の動作を説明する。

[0097] 評価値更新部 224 は、ステップ S304 の後、式 1～式 3 を用いて、m

eanValue、bestValue、kbestValue(k)を算出する。

[0098] 評価値更新部224は、ノードごとに次の計算式を用いてノードの総合評価値を得る。なお、 $w_{best}(n)$ 、 $w_{mean}(n)$ 、 $w_{kbest(k)}(n)$ 、 $\varepsilon_{best}(n)$ 、 $\varepsilon_{mean}(n)$ 、 $\varepsilon_{kbest(k)}(n)$ は、解空間木がN段のレベルを有するとした場合の、n段目における各指標best、mean、kbestに対する重みを表す係数である

[0099] [数7]

$$hValue(bestValue, meanValue, kbestValue(k)) = \frac{w_{best}(n) \left(\frac{1}{1 + \varepsilon_{best}(n)} \right) + w_{kbest(k)}(n) \left(\frac{1}{1 + \varepsilon_{kbest(k)}(n)} \right) + w_{mean}(n) \left(\frac{1}{1 + \varepsilon_{mean}(n)} \right)}{w_{best}(n) \left(\frac{1}{bestValue + \varepsilon_{best}(n)} \right) + w_{kbest(k)}(n) \left(\frac{1}{kbestValue(k) + \varepsilon_{kbest(k)}(n)} \right) + w_{mean}(n) \left(\frac{1}{meanValue + \varepsilon_{mean}(n)} \right)}$$

[0100] また、評価値更新部224は、以下のように、wやεの値を、ノード選択に用いる評価関数と、枝刈に用いる評価関数とで区別する。例えば、ノード選択に用いる評価関数では、

[0101] [数8]

$$\begin{aligned} w_{best}(n) &= \frac{n}{N} \\ w_{kbest(k)}(n) &= \min \left(\frac{n}{N}, 1 - \frac{n}{N} \right) \\ w_{mean}(n) &= \max \left(1 - \frac{2n}{N}, 0 \right) \end{aligned}$$

[0102] とする。枝刈に用いる評価関数では、

[0103] [数9]

$$\begin{aligned} w_{best}(n) &= 0.9 \frac{n}{N} \\ w_{kbest(k)}(n) &= 0.9 \left(1 - \frac{n}{N} \right) \\ w_{mean}(n) &= 0.1 \end{aligned}$$

[0104] とする。

[0105] 以上に説明したように、本実施形態では、第3の実施形態と同様の効果が得られるとともに、各ノードの探索木における深さにおいて、評価関数を変えることができる。それにより、ノードのシミュレーション回数を適切に増やすことができる。従って、より重要なノードのシミュレーション回数をより適切に増やすことができるので、MCTSなど、シミュレーションによって評価値を算出する探索方法の解の精度をさらに改善する可能性を高めることができる。

[0106] 実施形態5.

以下、本発明の第5の実施形態を図面を参照して説明する。

[0107] 第5の実施形態における最適化システムの構成は、第1の実施形態の構成と同様である。

[0108] しかし、最適化装置2の評価値更新部224は、枝刈に用いる評価関数を、解空間木の大きさを抑えることを目的とする枝刈に用いる評価関数と、指定された計算時間内に探索木を解空間木の底に到達させることを目的とする枝刈に用いる評価関数とに区別して、各ノードに設定する。本実施形態では、枝刈に用いる評価関数をmeanとし、解空間木の大きさを抑えることを目的とする枝刈に用いる評価関数をkbestとし、指定された計算時間内に探索木を解空間木の底に到達させることを目的とする枝刈に用いる評価関数をbestとする。

[0109] 第5の実施形態における計算部22の動作は、図3に示す第1の実施形態の動作と同様である。

[0110] しかし、ステップS305における評価値更新部224、およびステップS306における枝刈部225の動作が異なる。ここで、ステップS305、ステップS306における動作を説明する。

[0111] ステップS305において、評価値更新部224は、式1～式3を用いて、meanValue、bestValue、kbestValue(k)を算出する。

[0112] 評価値更新部224は、meanValue、kbestValue(k

）、`best Value`を、選択ノードとその上位のノードそれぞれで計算し、算出結果をもとにそれぞれのノードの評価値を更新する。

[0113] `mean Value`は、選択部221におけるノードの選択処理に用いられる。また、`k best Value (k)`は、枝刈部225における、探索木の大きさを抑えることを目的とする枝刈処理に用いられる。また、`best Value`は、枝刈部225における、指定された計算時間内に探索木を解空間木の底に到達させることを目的とする枝刈処理に用いられる。

[0114] ステップS306において、枝刈部225は、選択部221がノード選択に用いる評価関数とは異なる評価関数`k best Value (k)`または`best Value`で、各ノードを評価し、枝刈処理を実行する。このとき、枝刈部225は、実行する枝刈処理の目的に応じて、`k best Value (k)`または`best Value`のどちらを用いるかを決定する。すなわち、枝刈処理が、探索木の大きさを抑えることを目的とするものであるか、指定された計算時間内に探索木を解空間木の底に到達させることを目的とするものであるかにより決定する。

[0115] 以上に説明したように、本実施形態では、第2の実施形態と同様の効果が得られるとともに、枝刈に用いる評価関数を、枝刈の目的ごとに設定することができる。それにより、枝刈の目的に応じて、価値の低いノードをより適切に選択し、取り除くことができる。例えば、探索木の大きさを抑えることを目的とする枝刈においては、ある程度平均的な期待値(`mean`)に近いものを残すことで、まだ少ないシミュレーション回数においても枝刈で大きな間違いを犯すリスクを減らすことができる。計算時間内に探索木を解空間木の底に到達させることを目的とする枝刈は、各ノードにおいてシミュレーション回数がある程度の回数を超えた段階における枝刈であり、`best`を用いたとしてもリスクはかなり軽減することができると考えられる。従って、より適切にかつ効率的に、価値の低いノードを取り除くことができる。よって、実行可能なメモリ空間の中で、より価値の高いノードのシミュレーション回数をさらに増やすことができる。

[0116] なお、第5の実施形態では、枝刈部を1つ含む計算部を例にしたが、図5に示すように、計算部22は、探索木の大きさを抑えることを目的とする枝刈処理を行う第1の枝刈部2251と、指定された計算時間内に探索木を解空間木の底に到達させることを目的とする枝刈処理を行う第2の枝刈部2252との2つ枝刈部を含んでいてもよい。図5は、枝刈部を2つ含む最適化装置を備えた最適化システムの構成の一例を示すブロック図である。

[0117] 図6は、図5に示す最適化装置における計算部の動作を示すフローチャートである。

[0118] なお、ステップS601、ステップS602、ステップS606～S608、ステップS612の処理は、ステップS301、ステップS302、ステップS303～S305、ステップS308の処理と同様であるため、説明を省略する。

[0119] 選択部221で選択されたノードのプレイアウト回数が、事前に定められた条件を満たした時に（ステップS603のYes）、第1の枝刈部2251は、枝刈により探索木の大きさを抑える必要があるか否かを判定する（ステップS604）。枝刈が必要である場合には（ステップS604のYes）、第1の枝刈部2251が枝刈を実行する（ステップS605）。枝刈が必要でない場合には（ステップS604のNo）、拡大部222が探索木を1段下位のノードまで展開する。

[0120] 計算部22は、n段目における計算時間が事前に定められた終了条件を満たすまで、ステップS602～S608の処理を繰り返し実行する（ステップS609）。当該計算時間が事前に定められた終了条件を満たした場合には（ステップS609のYes）、計算部22は探索木が解空間木の底（N段目）に達したか否かを確認する（ステップS610）。探索木が解空間木の底に達していない場合には（ステップS610のNo）、第2の枝刈部2252が枝刈を実行する（ステップS611）。計算部22は、ステップS611の処理を実行した後、ステップS602の処理に戻る。探索木が解空間木の底に達した場合には（ステップS610のYes）、ステップS61

2の処理に移行する。

[0121] なお、各実施形態において、最適化装置がスケジューリング問題に適用される場合を例にしたが、本発明の適用範囲はその限りではない。本発明は、タスクを担当者に割り当てるスケジューリング問題などの組合せ最適化問題を中心に、最適化問題全般に適用することが可能である。また、最適化問題以外の解探索にも適用することが可能である。

[0122] 図7は、本発明による解探索装置の最小構成を示すブロック図である。図7に示すように、解探索装置（図1に示す最適化装置2に相当。）は、シミュレーションを用いた解探索において、探索木中の選択枝となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行する実行部101（図1に示す最適化装置2における選択部221、拡大部222およびシミュレーション部223に相当。）と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新する更新部102（図1に示す最適化装置2における評価値更新部224に相当。）と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す枝刈部103（図1に示す最適化装置2における枝刈部225に相当。）とを含む。

[0123] そのような構成によれば、MCTSにおいてシミュレーション結果を用いた枝刈をすることにより、適切に探索範囲を狭めることができる。それにより、メモリ使用量を増大させることなく、与えられた計算時間内で探索木を解空間木の底まで到達させることができ、確実に解を算出することができる。また、シミュレーション結果をもとに重要であると判断したノードのシミュレーション回数を増やすことができる。よって、MCTSなどのシミュレーションによって評価値を算出する探索方法の解の精度を改善することができる。

[0124] 上記の実施形態には、以下のような解探索装置も開示されている。

[0125] （1）枝刈部103は、指定された計算時間をもとに、枝刈を実行する時間

間隔を決定し、当該時間間隔にもとづいて、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す処理を実行する解探索装置。

[0126] そのような構成によれば、与えられた計算時間内で探索木を解空間木の底まで確実に到達させることができる。

[0127] (2) 枝刈部 103 は、解探索の時間の経過とともに、枝刈を実行する時間間隔を徐々に短くする解探索装置。

[0128] そのような構成によれば、計算時間をより必要とする根付近のノードに対して、より多くの計算時間を割り当てることができる。

[0129] (3) 枝刈部 103 は、探索木の大きさが一定の大きさを超えたときに、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す解探索装置。

[0130] そのような構成によれば、探索木の大きさを一定の大きさ以下に確実に抑えることができる。

[0131] (4) 枝刈部 103 は、時間間隔にもとづく枝刈と探索木の大きさにもとづく枝刈とで、別々の評価値を用いる解探索装置。

[0132] そのような構成によれば、枝刈の目的に応じて、価値の低いノードをより適切に選択し、取り除くことができる。従って、より適切にかつ効率的に、価値の低いノードを取り除くことができる。

[0133] (5) 枝刈部 103 は、実行部 101 がノードを選択するときに用いる評価値とは別の評価値にもとづいて、ノードを探索木から切り離すか否かを判定する解探索装置。

[0134] そのような構成によれば、枝刈において、価値の低いノードを適切に選択し、取り除くことができる。

[0135] (6) 更新部 102 は、シミュレーション結果をもとに、複数の評価関数を用いて、複数の評価関数ごとに評価値を算出し、算出した各評価値の調和平均を総合評価値として算出し、総合評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新し、枝刈部 103 は、総合評価値をもとにノードを探索木から切り離すか否かを判定する解探索装置。

- [0136] そのような構成によれば、調和平均の各評価値に対する重み係数を、ノードを選択するための評価関数と、枝刈に用いる評価関数とで区別することにより、枝刈において、価値の低いノードをより適切に選択し、取り除くことができる。
- [0137] (7) 更新部 102 は、探索木の深さに応じて、調和平均を算出する際の各評価値に対する重み係数を変更する解探索装置。
- [0138] そのような構成によれば、各ノードの探索木における深さにおいて、評価関数を変えることができる。それにより、ノードのシミュレーション回数を適切に増やすことができる。
- [0139] また、上記の実施形態の一部又は全部は、以下の付記のようにも記載されるが、以下に限られない。
- [0140] (付記 1) シミュレーションを用いた解探索において、探索木中の選択肢となるノードの中からシミュレーションの実行対象となるノードを選択し、選択されたノードからシミュレーションを実行する実行部 101 と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新する更新部 102 と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す枝刈部 103 とを含むことを特徴とする解探索装置。
- [0141] (付記 2) 枝刈部 103 は、指定された計算時間をもとに、枝刈を実行する時間間隔を決定し、当該時間間隔にもとづいて、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す処理を実行する付記 1 に記載の解探索装置。
- [0142] (付記 3) 枝刈部 103 は、解探索の時間の経過とともに、枝刈を実行する時間間隔を徐々に短くする付記 2 に記載の解探索装置。
- [0143] (付記 4) 枝刈部 103 は、探索木の大きさが一定の大きさを超えたときに、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す付記 1 から付記 3 のうちのいずれか 1 つに記載の解探索装置。
- [0144] (付記 5) 枝刈部 103 は、時間間隔にもとづく枝刈と探索木の大きさにも

とづく枝刈とで、別々の評価値を用いる付記 4 に記載の解探索装置。

[0145] (付記 6) 枝刈部 103 は、実行部 101 がノードを選択するときに用いる評価値とは別の評価値にもとづいて、ノードを探索木から切り離すか否かを判定する付記 1 から付記 5 のうちのいずれか 1 つに記載の解探索装置。

[0146] (付記 7) 更新部 102 は、シミュレーション結果をもとに、各ノードに割り当てられた、ノードを選択するための評価関数およびノードを探索木から切り離すための評価関数を用いて、それぞれの評価関数ごとに評価値を算出する付記 6 に記載の解探索装置。

[0147] そのような構成によれば、枝刈において、価値の低いノードをより適切に選択し、取り除くことができる。

[0148] (付記 8) 更新部 102 は、シミュレーション結果をもとに、各エッジに割り当てられた、ノードを選択するための評価関数と、各ノードに割り当てられた、ノードを探索木から切り離すための評価関数とを用いて、それぞれの評価関数ごとに評価値を算出する付記 6 に記載の解探索装置。

[0149] そのような構成によれば、評価関数をエッジに割り当てることが可能となり、本発明をツリーだけでなく、例えば、非循環有向グラフ (DAG: Directed Acyclic Graph) にも適用することが可能となる。

[0150] (付記 9) 更新部 102 は、シミュレーション結果をもとに、複数の評価関数を用いて、複数の評価関数ごとに評価値を算出し、算出した各評価値の調和平均を総合評価値として算出し、総合評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新し、枝刈部 103 は、総合評価値をもとにノードを探索木から切り離すか否かを判定する付記 1 から付記 8 のうちのいずれか 1 つに記載の解探索装置。

[0151] (付記 10) 更新部 102 は、探索木の深さに応じて、調和平均を算出する際の各評価値に対する重み係数を変更する付記 9 に記載の解探索装置。

[0152] この出願は、2013 年 1 月 25 日に提出された日本特許出願 2013-011628 を基礎とする優先権を主張し、その開示の全てをここに取り込

む。

[0153] 以上、実施形態を参照して本願発明を説明したが、本願発明は上記の実施形態に限定されるものではない。本願発明の構成や詳細には、本願発明のスコープ内で当業者が理解し得る様々な変更をすることができる。

符号の説明

- [0154]
- 1 ユーザ端末
 - 2 最適化装置
 - 1 1 操作部
 - 1 2 表示部
 - 2 1 G U I 部
 - 2 2 計算部
 - 2 3 記憶部
 - 1 0 1 実行部
 - 1 0 2 更新部
 - 1 0 3、2 2 5 枝刈部
 - 2 2 1 選択部
 - 2 2 2 拡大部
 - 2 2 3 シミュレーション部
 - 2 2 4 評価値更新部
 - 2 2 5 1 第 1 の枝刈部
 - 2 2 5 2 第 2 の枝刈部

請求の範囲

- [請求項1] シミュレーションを用いた解探索において、探索木中の選択肢となるノードの中からシミュレーションの実行対象となるノードを選択し、選択された前記ノードからシミュレーションを実行する実行部と、シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択された前記ノードおよびその上位ノードの評価値を更新する更新部と、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す枝刈部とを含むことを特徴とする解探索装置。
- [請求項2] 枝刈部は、指定された計算時間をもとに、枝刈を実行する時間間隔を決定し、当該時間間隔にもとづいて、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す処理を実行する請求項1に記載の解探索装置。
- [請求項3] 枝刈部は、解探索の時間の経過とともに、枝刈を実行する時間間隔を徐々に短くする請求項2に記載の解探索装置。
- [請求項4] 枝刈部は、探索木の大きさが一定の大きさを越えたときに、予め定められた基準を満たさない評価値を持つノードを探索木から切り離す請求項1から請求項3のうちのいずれか1項に記載の解探索装置。
- [請求項5] 枝刈部は、時間間隔にもとづく枝刈と探索木の大きさにもとづく枝刈とで、別々の評価値を用いる請求項4に記載の解探索装置。
- [請求項6] 枝刈部は、実行部がノードを選択するときに用いる評価値とは別の評価値にもとづいて、ノードを探索木から切り離すか否かを判定する請求項1から請求項5のうちのいずれか1項に記載の解探索装置。
- [請求項7] 更新部は、シミュレーション結果をもとに、複数の評価関数を用いて、前記複数の評価関数ごとに評価値を算出し、算出した各評価値の

調和平均を総合評価値として算出し、前記総合評価値をもとに、選択されたノードおよびその上位ノードの評価値を更新し、

枝刈部は、前記総合評価値をもとにノードを探索木から切り離すか否かを判定する

請求項 1 から請求項 6 のうちのいずれか 1 項に記載の解探索装置。

[請求項8] 更新部は、探索木の深さに応じて、調和平均を算出する際の各評価値に対する重み係数を変更する

請求項 7 に記載の解探索装置。

[請求項9] シミュレーションを用いた解探索において、探索木中の選択肢となるノードの中からシミュレーションの実行対象となるノードを選択し、選択された前記ノードからシミュレーションを実行し、

シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択された前記ノードおよびその上位ノードの評価値を更新し、

予め定められた基準を満たさない評価値を持つノードを探索木から切り離す

ことを特徴とする解探索方法。

[請求項10] コンピュータに、

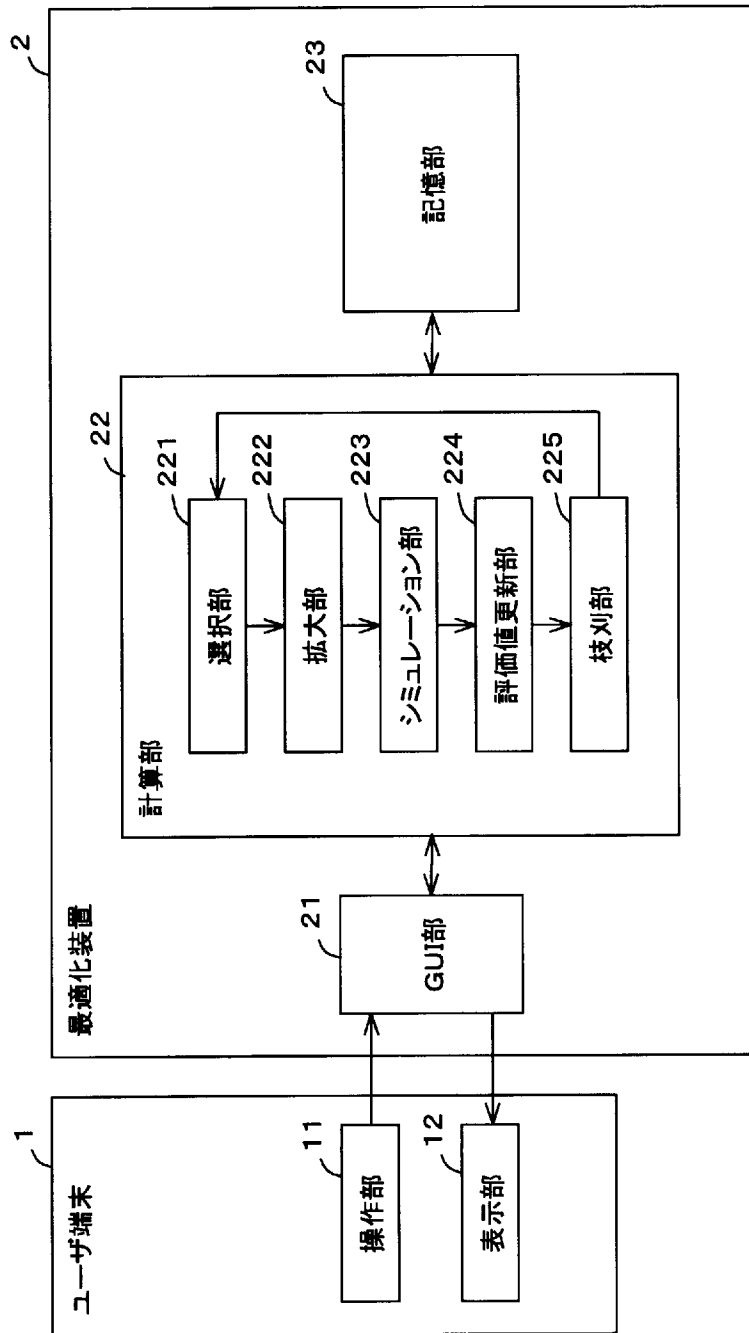
シミュレーションを用いた解探索において、探索木中の選択肢となるノードの中からシミュレーションの実行対象となるノードを選択し、選択された前記ノードからシミュレーションを実行する処理と、

シミュレーション結果をもとに、評価関数を用いて評価値を算出し、当該評価値をもとに、選択された前記ノードおよびその上位ノードの評価値を更新する処理と、

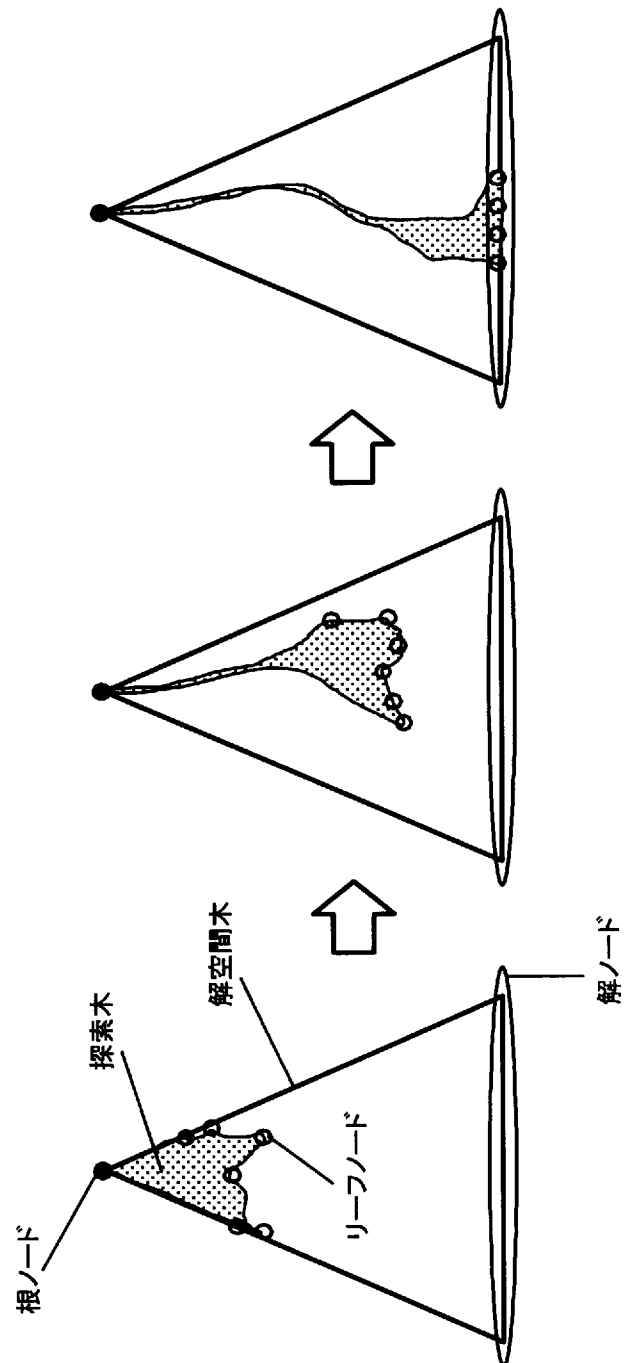
予め定められた基準を満たさない評価値を持つノードを探索木から切り離す処理とを

実行させるための解探索プログラム。

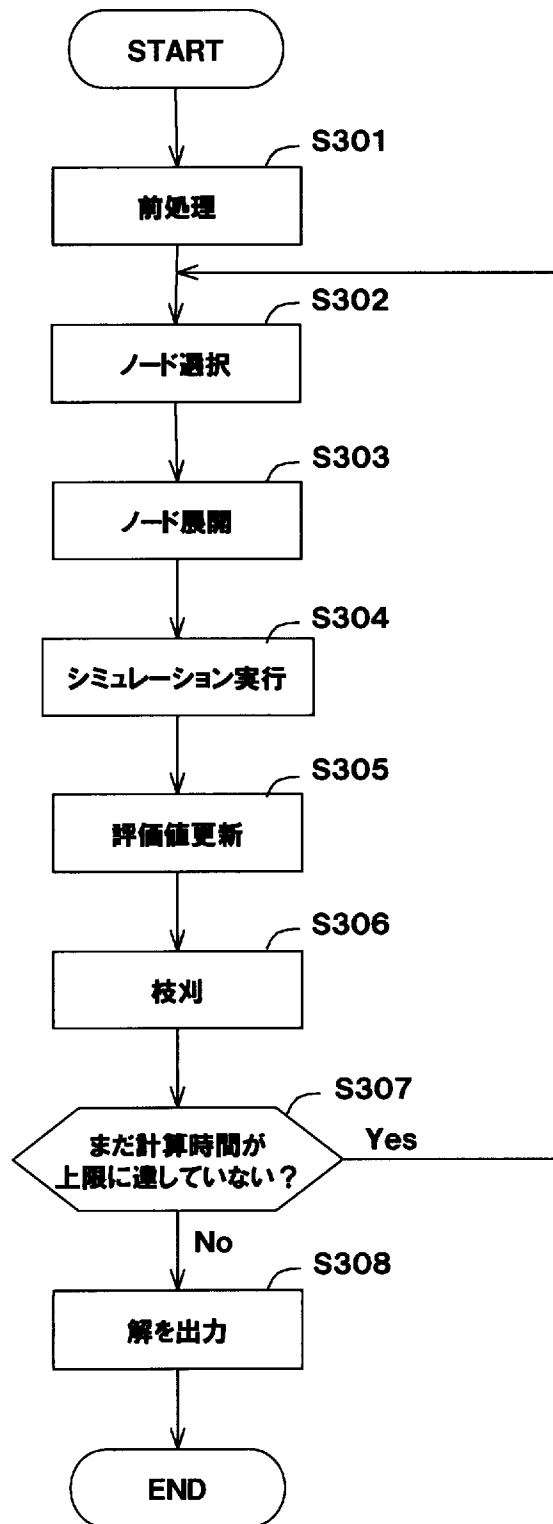
[図1]



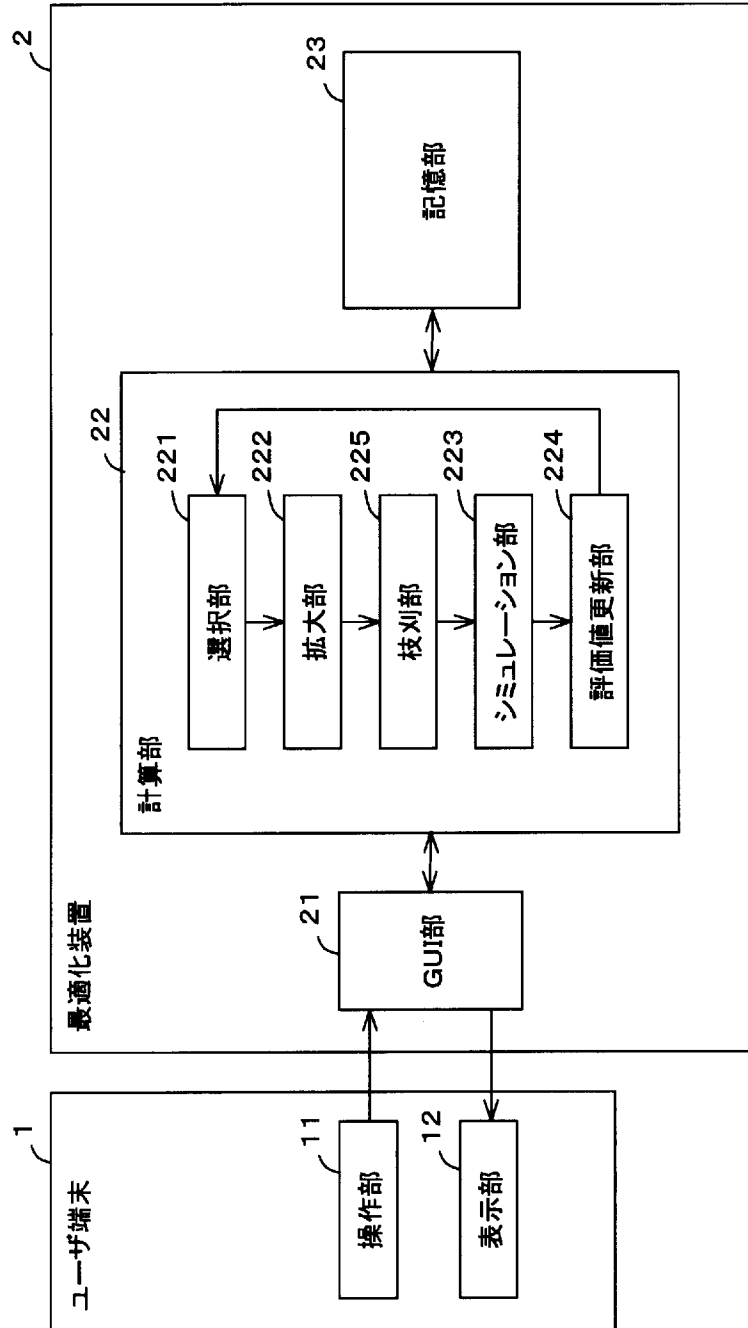
[図2]



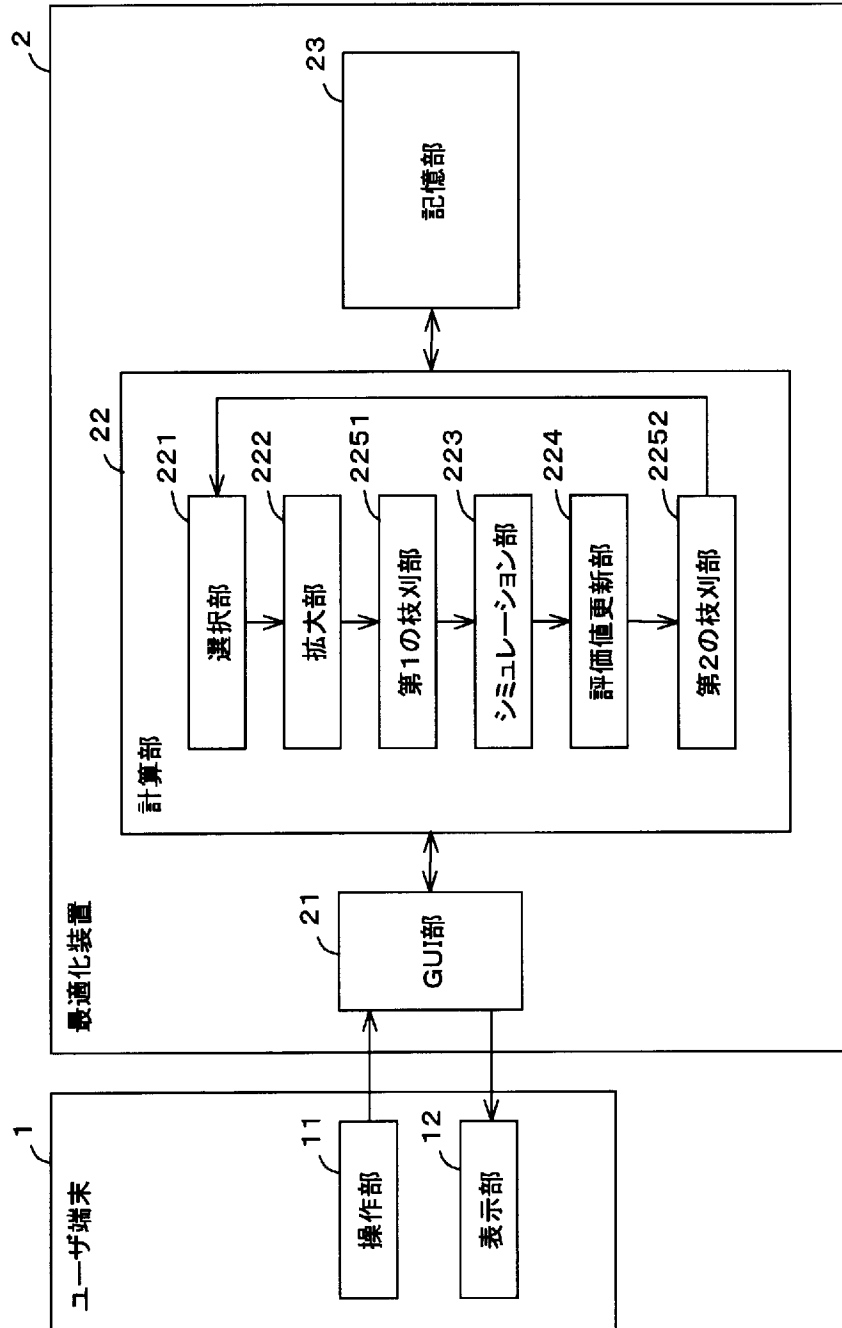
[図3]



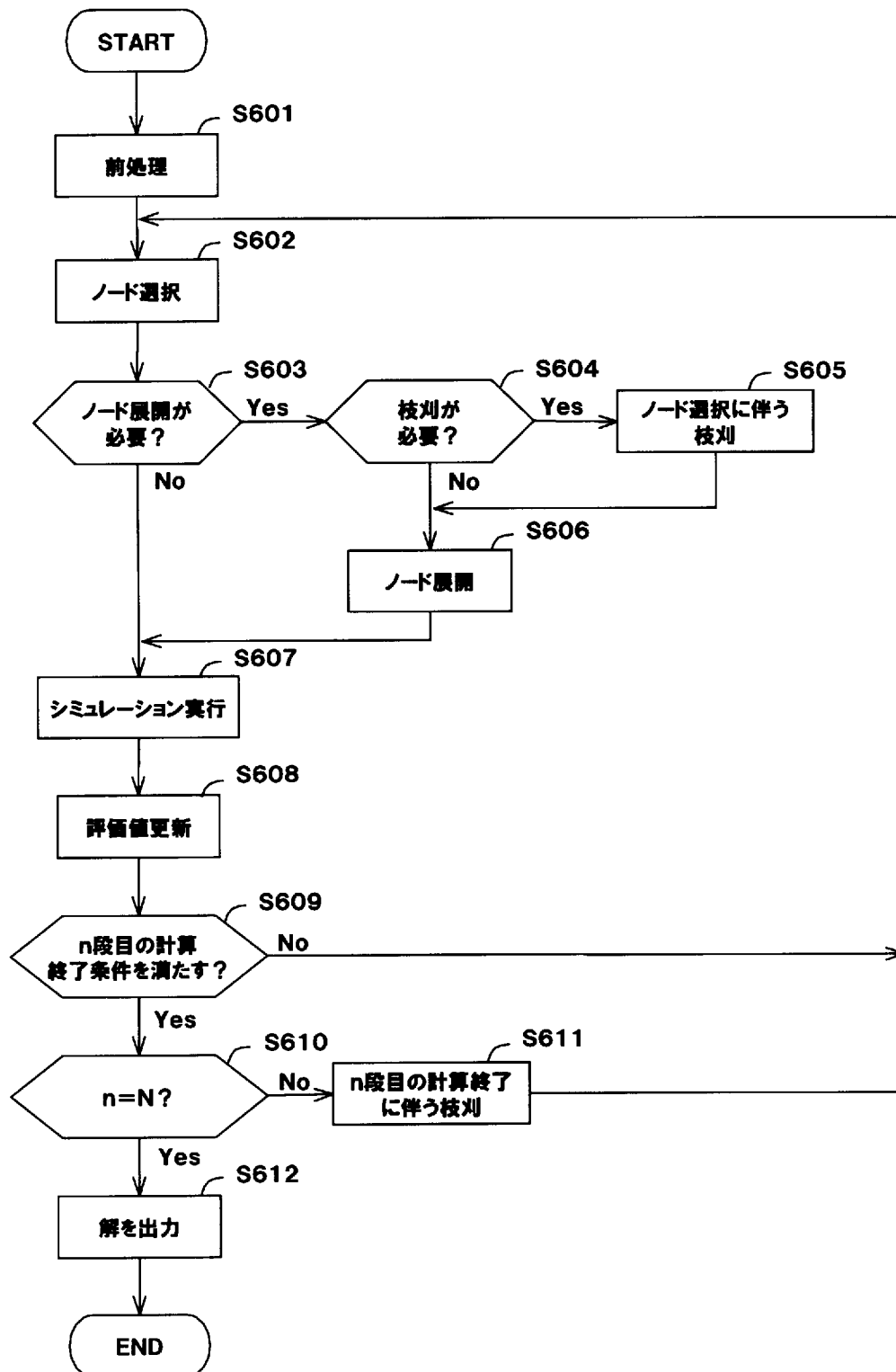
[図4]



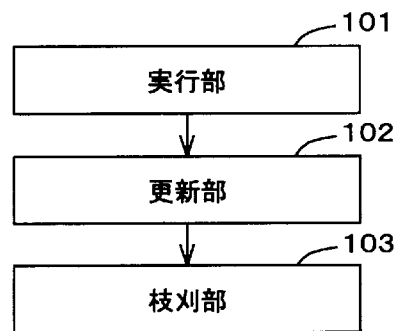
[図5]



[図6]



[図7]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/007673

A. CLASSIFICATION OF SUBJECT MATTER

G06N5/04(2006.01)i, G06F19/00(2011.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N5/04, G06N3/00, G06F19/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2014
Kokai Jitsuyo Shinan Koho	1971-2014	Toroku Jitsuyo Shinan Koho	1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JSTPlus (JDreamIII)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	Ryuhei KITAGAWA et al., "Pruning in UCT search based on limitedness of computational resource", Dai 13 Kai Game Programming Workshop 2008 Ronbunshu, 31 October 2008 (31.10.2008), vol. 2008, no.11, pages 46 to 53, particularly, page 46, right column, lines 12 to 23; page 47, left column, line 13 to page 49, left column, line 10	1, 6, 9, 10 2-5, 7, 8
Y	JP 2011-27910 A (KDDI Corp.), 10 February 2011 (10.02.2011), paragraph [0052] & US 2011/0022385 A1	2-5

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
11 March, 2014 (11.03.14)

Date of mailing of the international search report
18 March, 2014 (18.03.14)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/007673

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Kendai IJIRI et al., "A Method to Assess Suitability of Solution Space for Simulated Annealing", IPSJ SIG Notes, 05 March 2002 (05.03.2002), vol.2002, no.19, pages 33 to 36, particularly, page 35, left column, lines 24 to 38	7, 8
A	Cameron B. BROWNE et al., A Survey of Monte Carlo Tree Search Methods, IEEE Transactions on Computational Intelligence and AI in Games, 2012.03, Vol.4, No.1, pp.1-43 particularly, passages III.A, III.C	1-10

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06N5/04(2006.01)i, G06F19/00(2011.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06N5/04, G06N3/00, G06F19/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 4 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 4 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 4 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

JSTPlus(JDreamIII)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X Y	北川 竜平 外 2 名, 投入計算量の有限性に基づく U C T 探索の枝刈り, 第 1 3 回 ゲーム・プログラミング ワークショップ 2 0 0 8 論文集, 2008. 10. 31, Vol. 2008, No. 11, pp. 46-53 特に, 第 4 6 ページ右欄第 1 2 行ー同欄第 2 3 行、第 4 7 ページ左欄第 1 3 行ー第 4 9 ページ左欄第 1 0 行の記載を参照。	1, 6, 9, 10 2-5, 7, 8
Y	JP 2011-27910 A (K D D I 株式会社) 2011. 02. 10, 段落【0 0 5 2】 & US 2011/0022385 A1	2-5

☒ C 欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の 1 以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

1 1 . 0 3 . 2 0 1 4

国際調査報告の発送日

1 8 . 0 3 . 2 0 1 4

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）
郵便番号 1 0 0 - 8 9 1 5
東京都千代田区霞が関三丁目 4 番 3 号

特許庁審査官（権限のある職員）

稲垣 良一

5 B

4 0 5 5

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 4 4

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
Y	井尻 堅大 外 1 名, Simulated Annealing 法と解空間との適性の評価法, 情報処理学会研究報告, 2002. 03. 05, Vol. 2002, No. 19, pp. 33-36 特に、第 3 5 ページ左欄第 2 4 行ー同欄第 3 8 行の記載を参照。	7, 8
A	Cameron B. BROWNE et al., A Survey of Monte Carlo Tree Search Methods, IEEE Transactions on Computational Intelligence and AI in Games, 2012. 03, Vol. 4, No. 1, pp. 1-43 特に、第 III. A 節, 第 III. C 節の記載を参照。	1-10