

LC-007

アントコロニー最適化アルゴリズム専用ハードウェア

Dedicated Hardware for Ant Colony Optimization

原田隆史†

吉川雅弥‡

寺井秀一†

Takashi Harada†

Masaya Yoshikawa‡

Hidekazu Terai†

1. はじめに

組合せ最適化問題の多くは、現実的な計算時間で最適解を求めることがきわめて困難であることが知られている。しかし、最適解でなくても十分精度の高い解であれば実用上問題がない場合が多い。そこで現実的な計算時間で良質な解を求めるためにメタヒューリスティック解法が用いられる。近年、このメタヒューリスティック解法の中で、アリの採餌行動を模倣したアントコロニー最適化手法 (Ant Colony Optimization: 以下, ACO) [1]が注目されている。このACOは、組み合わせ最適化問題に対する強力な解探索能力を有するが、繰り返し処理が多いため処理時間の問題が内在する。そこで本研究では、高速なACOの処理を実現する専用ハードウェアを提案する。また、ソフトウェアとの比較実験により提案ハードウェアの有効性を確認した。

2. これまでの関連研究

これまでにACOに関連する研究がいくつかなされている。まず、Marco Dorigo, Luca Maria Gambardella[2]らは、巡回セールスマン問題を対象としてACOとシュミレーティッドアニーリング法との比較を行い、ACOの有効性を示している。また、Vittorio Maniezzo, Alberto Colomi[3]らは、ジョブショップスケジューリング問題を対象として、タブサーチ法との比較によってACOの有効性を検証している。このように、これまでの発表されているACOに関連する研究では、他のメタヒューリスティック解法と比較して良好な結果を残している。しかしながら、これまでの研究は、ACOをソフトウェアで実現したものがほとんど全てであり、本研究で提案する専用ハードウェアによるACOの実現は、新しいアプローチである。

3. アーキテクチャメソッドロジー

3.1 ハードウェアアルゴリズム

専用ハードウェアでACOを実現するためには、まず基準となるアルゴリズムの特性を把握する事が重要となる。まず、ソフトウェアでのACOアルゴリズムには繰り返し処理が多いため処理時間が多くかかるという問題点がある。そこで提案アーキテクチャでは、ソフトウェアでのアルゴリズムの問題点を考慮し、アルゴリズムの特徴を残しつつ、ハードウェア用にチューニングし実装することでこの問題を克服した。この提案アルゴリズムには以下の2つの特徴がある。

一つ目の特徴は、都市選択部分と局所更新部分を一つのプロセッサエレメント(processor element 以下 PE)とする事で、PEの数に応じた台数効果の高速性を実現した。

二つ目の特徴は、各PEが独立して経路探索を行うため、各PEが生成した経路を統合する必要がある。この統合処理をフェロモンの更新回路に組み込む事で処理の並列化を実現した。以上を踏まえた提案アルゴリズムのフローチャートを図1に示す。

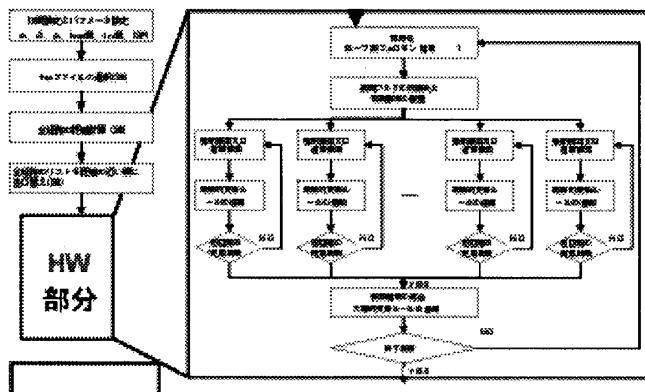


図1 提案アルゴリズムのフローチャート

3.2 IDの構成

次に、多くのSRAMを複数のアリエージェント全員が共有しているため、データ処理の競合が起こるといった問題点がある。そこで、データにIDをつけ管理することで解決した。また、設計仕様を決めるためにソフトウェアで予備実験を行った。フェロモン処理で最も多く行われる浮動小数点演算の精度と処理時間の観点からビット幅を24ビットにした。また、データを読み込むか、書き込むかを識別するために、上位1ビットをRead Write IDとし、次の3ビットはデータの種別 (SRAM)を識別するためのMemory IDとし、残りの13ビットは、データのアドレスとなっている。IDの構成を図2に示す。

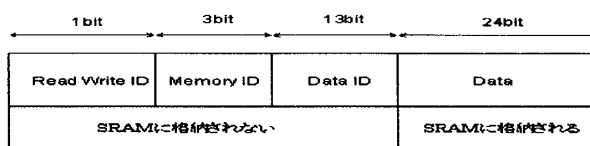


図2 IDの構成

4. 回路構成

今回設計した全体の回路はインターフェース回路と、提案アーキテクチャを搭載した演算部分で構成する。大きく分けてインターフェース回路、制御回路であるメインコントローラ回路、都市選択及び、局所更新を行うPE、各PEの演算結果を統合及び、大域更新を行う大域更新回路、終了判断を行う判断回路で構成されている。回路ブロック図を図3に示す。

†立命館大学院 理工学研究科 Ritsumeikan University
‡名城大学 理工学部 情報工学科 Meijo University

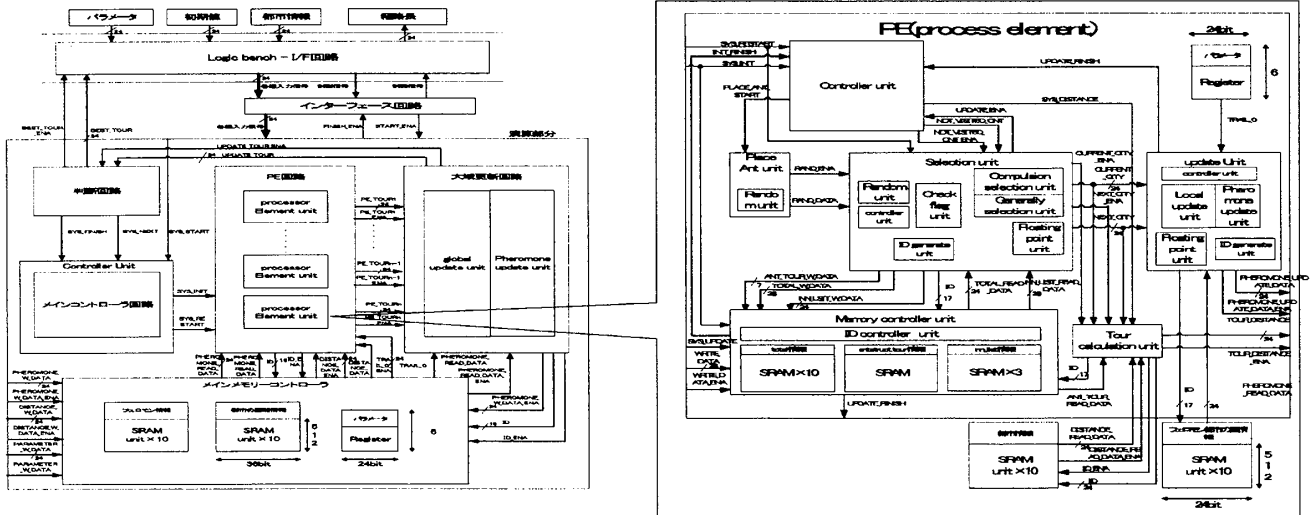


図3 回路ブロック図

5. 実験結果

作成した回路の性能及び、必要なゲート数の見積もりをするため本研究では、Windows XP SP2, Pentium4 3.0 GHz 上で、ISE8.2 を使用し、各回路の回路規模を見積もった。PE の数による全体回路規模の見積もりを表 1 に示す。

表 1 PE の数による全体回路規模の見積もり

	PE1	PE2	PE5	PE10
slices	9794	14744	29595	54347
slices FF	3466	5427	11308	21110
4 inputs LUTs	18860	28288	56548	103648
クリティカルパス[ns]	22.853	22.853	22.853	22.853
動作周波数 [MHz]	43.759	43.759	43.759	43.759
RAMB16s	41	58	109	194

次に SW と HW の ACO を比較するために、Windows XPSP2, Pentium4 3.6GHz 上で検証を行った。組合せ問題の基礎となる巡回セールスマン問題で、TSPLIB[4]に公開されている kroA100 問題を用いた。

設定パラメータは、アリエージェントの数を 100, α を 7.0, β を 9.0, ρ を 0.5, 最大ステップ数は 80000 回、試行実行の回数は 10 回とした。

KroA100 問題の SW と HW の速度比較結果を表 2 に示す。

表 2 kroA100 都市問題の SW と HW の速度比較結果

PE 数	SW(s)	HW(s)	比較
1	2554	1230.744409	2.075166852
2	2554	620.3444088	4.117067816
5	2554	254.1044088	10.05098657
10	2554	132.0244088	19.34490769

表 1 より、ACO の回路規模及び SRAM 数は PE の並列数を変化させることで増減することが確認できた。これらは並列数を減らすことで縮小できるが、そうした場合は並列数を減らすことにより実行時間に影響を及ぼす。FPGA への実装においては回路規模および SRAM 数と実行時間とのトレードオフを考える必要がある。また、表 2 より、並列数がいずれの場合もハードウェアの処理時間がソフトウェアの処理時間より短く、ハードウェア化の有効性が確認できた。これは、ソフトウェアでボトルネックである探索処理を複数で同時探索させているためであり、PE を多くする程ハードウェアが有効である。さらに、並列数がいずれの場合も最適解に達しており、提案アーキテクチャが正常に動作していることが確認できた。

6. おわりに

本論文では、巡回セールスマン問題を対象とした ACO の専用ハードウェアを提案した。提案したハードウェアでは、PE を並列処理することで高速処理を実現した。また、データを ID 管理することで、効率の良い SRAM へのアクセスを可能とした。さらに、ベンチマークデータによるステップ数評価実験では、ソフトウェア処理に比べて、PE を 10 並列にした時、最大 19.3 倍の高速化を実現した。今後の研究の課題としては、動的問題への対応や、さらなる高速処理のための拡張機能の検討などが挙げられる。

7. 参考文献

- [1] 大内東, 山本雅人, 川村秀憲, 柴肇一, 高柳俊明, 當間俊明, 遠藤聡志. 「生命複雑系からの計算パラダイム」 森北出版 pp.1-76, Aug. 2003
- [2] M. Dorigo, L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," IEEE Trans. Evolutionary Computation, Vol. 1, No. 1, pp.53-66, 1997
- [3] Alberto Colomi, Marco Dorigo, and Vittorio Maniezzo. In proceedings European Conference on Artificial Life, pages 134-142. MIT Press, 1991.
- [4] <http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSPLIB95/index.html>