

Title	計算幾何学問題に対するメモリ制約付きアルゴリズム
Author(s)	小長谷, 松雄
Citation	
Issue Date	2012-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/10416
Rights	
Description	Supervisor:浅野哲夫, 情報科学研究科, 修士

計算幾何学の問題に対するメモリ制約付きアルゴリズム

小長谷松雄 (1010022)

北陸先端科学技術大学院大学 情報科学研究科

2012 年 2 月 6 日

キーワード: アルゴリズム, 定数領域アルゴリズム, メモリ制約付きアルゴリズム, 計算幾何学, 作業領域.

メモリ制約付きアルゴリズムとは, できるだけ少ない作業領域の使用を考慮して設計されたアルゴリズムである. 作業領域が一番少ない場合は定数領域アルゴリズムと呼ばれ, $O(1)$ 領域だけの使用が許される. ここで, 入力配列は作業領域として考えないとする. 計算複雑性理論の分野では, 対数記憶領域アルゴリズムとよばれ 30 年以上前から研究されている. しかし, 本稿で示す定数領域アルゴリズムは理論だけでなく, 実装のしやすさについても重点を置いている.

作業領域が定数個の変数しか利用できない状況は非常に厳しい制限である. $O(s)$ の作業領域が確保できるならば, 効率よく問題を解くアルゴリズムを設計できるだろうか. 即ち, 記憶領域と計算時間のトレードオフをもつアルゴリズムの設計についても考える. s は領域のサイズを決定するパラメータを表す. トレードオフがあるアルゴリズムを用いることで, アルゴリズムが占有しうる領域のサイズを自由に設定できる. 一般に, 大きな作業領域を用いれば, 時間計算量は減少し, 計算時間を犠牲にすることで使用メモリは節約される.

本稿を通して, 入力データは読み出し専用の配列に格納されていると仮定する. この配列では, 配列内の要素の削除や並び替えが禁止されている. ある入力データが異なるアルゴリズムで並列に処理される環境では, 一方のアルゴリズムが, 入力要素を削除したり並び替えたりできる状況は望ましくない.

本稿では, 2 つの計算幾何学の問題に対するメモリ制約付きアルゴリズムを設計する. 一つめとして, 最遠点ボロノイ図の描画を行う定数領域アルゴリズムを設計する. このアルゴリズムは, n 点の入力に対して, $O(n^2)$ 時間で入力点に対応する最遠点ボロノイ図を描画する. ボロノイ図を描画するとは, 図を構成する頂点や辺を全て列挙するという意味がある. 列挙されたボロノイ頂点や辺は別の計算幾何の問題を解く方法に応用できる. 既存の手法は, まず, 入力に対する最遠点ボロノイ図を前もって計算し, 計算された頂点や辺を二重連結辺リストとして $O(n)$ の記憶領域に保持する. そして, リストの全ての辺や

頂点を $O(n)$ 時間で列挙することで、最小包含円を発見する。ただし、二重連結辺リストの構築に $O(n \log n)$ 時間を要する。

次に、平面に与えられた n 本の直線分に対し、交差する全ての線分対を列挙するためのメモリ制約付きアルゴリズムを考える。この問題では、計算時間と作業領域のトレードオフがあるアルゴリズムを設計する。

直線分交差の問題は計算幾何学で最も基本的な問題の一つであり、数多くのアルゴリズムが提案されている。Bentley と Ottman のアルゴリズムは平面走査法で、 $O((n+K) \log n)$ 時間で $O(K)$ の作業領域を実現した。その後 Balaban は、 $O(n)$ の作業領域で、 $O(n \log n + K)$ 時間で計算するアルゴリズムを提案した。 K は交差する直線分対の数を表す。これらのアルゴリズムは、入力配列も作業領域として用いている。また、作業領域をできるだけ減らした手法の研究もされている。Bentley と Ottman の平面走査法は、探索木を保持するために入力配列とは別に $\Omega(n)$ の作業領域を要する。Chan と Chen は、この作業領域を $O(\log^2 n)$ に改善し、 $O((n+K) \log^2 n)$ の計算時間を得た。一方、力任せの方法でも、定数領域で $O(n^2)$ 時間で全ての直線分交差を列挙できる。

本稿では、 $O(s)$ の作業領域だけを用いて、直線分の集合 S の交差する全ての線分対を求めるアルゴリズムを提案する。まずは S をグループ $S_1, \dots, S_{\lceil n/s \rceil}$ に分割して、各々のグループにおける線分交差を求める。そして、異なるグループ同士の交差線分対を求める。しかし、このアルゴリズムでは一つの交点を何度も検出する問題が起こる。

一つの交差発見をちょうど一回にするために、次のように改善する。 S を分割した後、 S_i に対してデータ構造を構築する。そして、このデータ構造を用いて、 $s \in S \setminus S_i$ と交差する S_i の線分を発見する。このデータ構造は、Chazell, Sharir, Welzl により提案されたので CSW のデータ構造と呼ばれており、様々な高度な技巧を用いたデータ構造である。

CSW のデータ構造を用いた手法は非常に複雑である。より単純な交差列挙のアルゴリズムを考えるために、入力線分を水平線分と垂直線分のみと仮定する。このとき、 $O(s)$ の作業領域を用いた簡単な平面走査法で、効率の良いトレードオフをもつアルゴリズムを設計できる。