

Title	補助目的関数を用いたSAに基づく矩形パッキングの収束性改善
Author(s)	小川, 真一
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1934
Rights	
Description	Supervisor:金子 峰雄, 情報科学研究科, 修士

修士論文

補助目的関数を用いたSAに基づく
矩形パッキングの収束性改善

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

小川 真一

2005年3月

修士論文

補助目的関数を用いたSAに基づく
矩形パッキングの収束性改善

指導教官 金子峰雄 教授

審査委員主査 金子峰雄 教授
審査委員 宮地充子 助教授
審査委員 平石邦彦 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

210015 小川 真一

提出年月: 2005 年 2 月

概要

VLSI (Very Large Scale Integrated circuit) チップレイアウト設計における部分問題の 1 つとして、モジュールの配置位置を決定する配置問題がある。配置問題は、各モジュールに対して 2 次元配置座標を与える問題であり、モジュール数を n とした時、その解空間は \mathbb{R}^{2n} である。村田らによって提案された Sequence-Pair はモジュール同士の相対位置関係を表現するもので、この相対位置の情報のみを制約としてモジュールをパッキングすることで対応するモジュール配置を得ようとするものであり、 $(n!)^2$ のサイズの有限解空間を持つ。

この有限の解空間を探索する有効な最適化手法の一つとして、Simulated Annealing (SA) 法が挙げられるが、集積回路の微細化技術の進歩と、1 チップ上に実装される回路規模の増大により、実際の問題に SA を適用して良好な解を得るためには膨大な計算時間を要する。このため、SA を高速化するための手法が求められている。

通常の矩形パッキングは全ての矩形を囲む最小矩形 (Bounding-Box) の面積にて解が評価されるため、一般に一つの解に対して評価値の等しい隣接解が多数存在する。それらの隣接解は SA の性質から無定見に受理されることになる。しかし面積が縮小するためには Bounding-Box の幅と高さを決定するモジュールの並び (critical path) が解消されなければならない。評価値の等しい隣接解の中でも、critical path が解消されやすい解は、critical path が解消されにくい解と比べて少ない変更回数で面積縮小に至ることができるため、同じ面積を持つ隣接解同士であっても critical path の解消されやすさを評価し解の優劣をつけることで、探索の効率化が期待される。

本研究ではこの考え方を具体化するために、critical path の解消されやすさを評価するためのカット次数と呼ぶ補助関数を導入し、カット次数を評価することによって、Bounding-Box の面積が同じ配置の中でも critical path が解消されやすい解へと探索を進める方法を検討する。

カット次数は、与えられたレイアウトにおいて現在存在する同一方向のすべての critical path が解消されるために、変更の必要があるモジュールの最小個数である。これを水平方向、垂直方向ともに求めることによって、面積縮小に至るまでに最低限必要な隣接解生成操作の繰り返し回数を評価できる。カット次数の算出は、次のように行なわれる。水平方向 (垂直方向) の critical path について、Bounding-Box の左右端 (上下端) をそれぞれ source(s) と sink(t)、critical path 上のモジュールを頂点、頂点と対応するモジュール間の相対位置関係を枝とするグラフ (critical path graph) を構成する。次にこのように構成された critical path graph において、頂点の中を流すことのできるフローのキャパシティを 1 とする s から t への最大フローを求める。このとき、最大フロー-最小カット定理より、この最大フローの値がカット次数となる。

カット次数を評価することで、従来の SA では無定見に受理されていた、critical path が解消されやすい解から解消されにくい解への変更を抑制し、面積縮小に至るまでの探索

回数を減らして探索を効率化することができる。これによって、カット次数を計算する時間を差し引いても、内部ループ回数を削減することで従来の SA に比べて高速化を図ることができる。

また、critical path 上のモジュール以外のモジュールを 2 個選んで変更を加えても、現存する critical path の解消にはつながらない。そこで隣接解生成操作で変更されるモジュールの中で、必ず 1 個は critical path 上のモジュールから選ばれるように改良を加えた。

提案手法を C 言語を用いて計算機上に実装し、実験によって従来の SA との比較を行い、提案手法の特徴および有効性を考察した。実験では、面積評価とカット次数評価の 2 段階で受理不受理を判定する方法と、カット次数と面積を組み合わせた 1 つの評価関数について受理不受理を判定する方法の比較、さらにカット次数を用いて 2 段階で受理不受理を判定する方法において、カット次数を評価する 2 段階目の温度スケジュールを変更した場合の比較、水平、垂直のカット次数の最小値を評価した場合と、水平、垂直のカット次数の和を評価した場合の比較を行ない、カット次数を有効に活用する方法を模索した。

本研究では、Bounding-Box の面積だけを評価する矩形パッキング問題を中心に考えたが、実際の VLSI チップレイアウト設計では配線長も最適化の対象となる。面積の最小化と配線長の最小化を同時に考える、実際のレイアウト設計における補助目的関数の導入と SA の効率化は、今後の課題である。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	2
第2章	準備	3
2.1	Sequence-Pair	3
2.2	Simulated Annealing(SA) 法	4
第3章	提案手法	6
3.1	critical path	6
3.2	カット次数	7
3.3	カット次数の SA への組み込み	9
第4章	実験と考察	12
4.1	準備	12
4.2	実験	13
4.2.1	実験1: カット次数を評価する2段階目の温度パラメーターに関する実験と比較	13
4.2.2	実験2: 従来のSAと、カット次数を用いたSAとの比較	13
4.2.3	実験3: 縦横のカット次数の最小値を評価した時と、和を評価した時の比較	13
4.2.4	実験4: 面積、カット次数の2段階で評価した時と、面積とカット次数を組み合わせた関数を1段階で評価した時の比較	13
4.2.5	実験5: カット次数、モジュール選択ありでSAの高速化を行なった場合と、従来のSAとの比較	14
4.2.6	実験6: ベンチマーク ami49 を用いた場合での比較	14
4.3	実験結果の考察	14
4.3.1	実験1: カット次数を評価する2段階目の温度パラメーターに関する考察	14
4.3.2	実験2: 従来のSAと、カット次数を用いたSAにおける実験結果の考察	14

4.3.3	実験 3 : 縦横のカット次数の最小値を評価した時と、和を評価した時の比較に関する考察	15
4.3.4	実験 4 : 面積、カット次数の 2 段階で評価した時と、面積とカット次数を組み合わせた関数を 1 段階で評価した時の比較に関する考察	15
4.3.5	実験 5 : カット次数、モジュール選択ありで SA の高速化を行なった場合と、従来の SA との比較に関する考察	16
4.3.6	実験 6 : ベンチマーク ami49 を用いた場合での実験結果に関する考察	16
第 5 章	結論	30
5.1	まとめ	30
5.2	今後の課題	30
第 6 章	謝辞	31

第1章 はじめに

1.1 背景

VLSI チップレイアウト設計における部分問題の1つとして、各モジュールに対して2次元配置座標を与える配置問題がある。モジュール数を n とした時、2次元配置問題の解空間は \mathbb{R}^{2n} であり、最適化のためにはこれを有限の解空間にする必要がある。そこで村田ら [1] は、矩形配置領域の面積最小化を目的とする時は、ある限定された配置 (具体的にはモジュールの左下詰め配置) のみを考えれば十分であることに注目し、 $(n!)^2$ のサイズの有限解空間を持つ Sequence-Pair と呼ばれる配置表現手法が提案された。Sequence-Pair はモジュール同士の相対位置関係を表現するもので、この相対位置の情報のみを制約としてモジュールをパッキングすることで対応するモジュール配置を得ることが可能である。

この有限の解空間を探索する最適化手法として、

1. モジュール間の結合の強さを指標としてクラスター化を行なう構成的手法
2. モジュール間の結合を引力、モジュール同士の重みを斥力として、力学的に平衡点を求める手法
3. Simulated Annealing(SA) 法、Genetic Algorithm(GA) などの確率的手法

などの手法がある。

このなかで、Simulated Annealing(SA) 法は様々な最適化目標に柔軟に対応でき、また比較的実装が容易であることから、2次元配置問題の解探索をする際にはよく用いられる手法である。しかしながら、近年の集積回路の微細化技術の進歩と、それに伴う1チップ上に実装される回路規模の増大により、実際の問題に SA を適用して良好な解を得るためには膨大な計算時間を要するため、SA を高速化する手法が求められている。

こうした問題に対して、異なる隣接解生成方法によって張られる解空間を、理論的に、または見ただ目で、実際の計算時間を調べることで解析 [2]、変更するモジュールから計算される値にしたがって、異なる隣接解生成方法を選択する方法 [3]、探索が進み温度が低くなった時に、受理される可能性の低い隣接解への遷移を制限する方法 [4][5] などの従来研究がある。

1.2 目的

通常の矩形パッキングは全ての矩形を囲む最小矩形 (Bounding-Box) の面積にて解が評価されるため、モジュールの配置が変化しても面積が変わらない、評価の等しい隣接解が多数存在する。それらの隣接解は、面積が良くなるか同じ場合には必ず受理されるという SA の性質から無定見に受理され、解は必ず変更されることになる。

本研究では、面積が縮小するためには Bounding-Box の幅と高さを決定するモジュールの並び (critical path) が解消されなければならないということ、さらに面積の等しい隣接解の中でも critical path が解消されやすい解と、critical path が解消されにくい解が存在することに注目した。critical path が解消されやすい解は、critical path が解消されにくい解と比べて少ない変更回数で面積縮小に至ることができるため、良い解であると言える。このように、同じ面積を持つ隣接解同士であっても、解の優劣をつけることで探索の効率化が期待される。

そこで本研究では、critical path の解消されやすさを評価するために、カット次数と呼ぶ補助関数を導入した。カット次数を SA の中で評価することによって、面積が等しい隣接解であっても critical path が解消されにくい解から critical path が解消されやすい解へと探索を進める方法を検討する。

第2章 準備

2.1 Sequence-Pair

Sequence-Pair は、モジュール同士の相対位置関係を表現するもので、矩形名の順列 $\Gamma+$ と $\Gamma-$ の対によって表される。Sequence-Pair では、 $\Gamma+$ と $\Gamma-$ でともに矩形 a が矩形 b の前にある時、つまり

$$\Gamma + (..a..b..), \Gamma - (..a..b..)$$

の時、矩形 a は矩形 b の左にあることを意味する。 $\Gamma+$ では矩形 a が矩形 b の前にあり、 $\Gamma-$ では矩形 a が矩形 b の後にある時、つまり

$$\Gamma + (..a..b..), \Gamma - (..b..a..)$$

の時、矩形 a は矩形 b の上にあることを意味する。図 2.1 に $\Gamma+(ecadfb), \Gamma-(fcbead)$ の時の左下詰めパッキング例を示す。

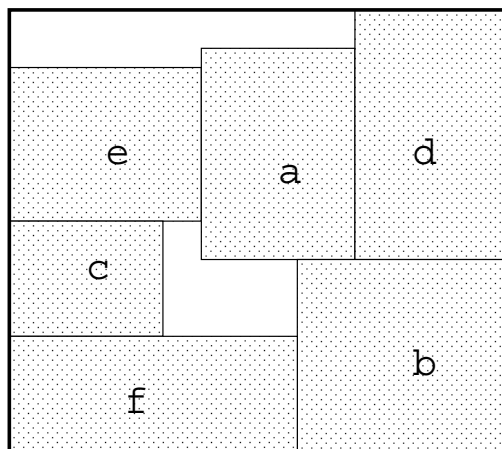


図 2.1: 左下詰めパッキングの例

n 個の矩形からなる順列対は $(n!)^2$ 通り存在し (回転を許すと $(n!)^2 \times 2^n$)、これを全探索により厳密解を求めるのは困難である。そこで多くの場合、Simulated Annealing(SA)法を用いて解空間を探索し、良好な解を求める。

2.2 Simulated Annealing(SA)法

SAでは、まず解の集合を定義して各解についての評価値を与え、集合中の1つの解から別の解を生成する隣接解生成操作を定義することで、解空間を構成する。次に、解空間の中から任意に選択した初期解を出発点とし、隣接解生成操作を繰り返すことで、最良の解を探索することを目指す。隣接解生成操作によって得られた解は、評価が同じか良化した場合は必ず受理され、悪化した場合は確率的に受理、不受理を決める。これにより探索が局所最適に陥ることを防ぎ、さらなる良解に向かって探索を進めることができる。

現在の解の評価関数を $f(s)$ 、生成した隣接解の評価関数を $f'(s)$ とすると、改善量 Δc は、

$$\Delta c = f(s) - f'(s)$$

で表される。

$$\Delta c \geq 0$$

ならば、解は良化しているので無条件に隣接解を受理し、解を更新する。

$$\Delta c < 0$$

ならば、確率 P で隣接解を受理することになり、解が受理される確率 P は、温度パラメータを T とした時、

$$P = \exp(\Delta c/T)$$

で表される。温度パラメータ T を十分高い温度から始め、探索を進めるしたがって T を徐々に下げていくことによって、評価関数が悪化する解への変更を抑えていくことができる。2次元配置問題をSAで解く際の隣接解生成操作は、ここでは

- 全交換：選択された2つのモジュールについて、 Γ_+ 、 Γ_- の両方で入れ替える。
- Γ_+ 交換：選択された2つのモジュールについて、 Γ_+ のみ入れ替える。
- 回転：選択された1つのモジュールを90度回転する。

の3種類の組み合わせを用いるものとする [2]。図 2.2 は、一般的なSAのアルゴリズムの一例である。

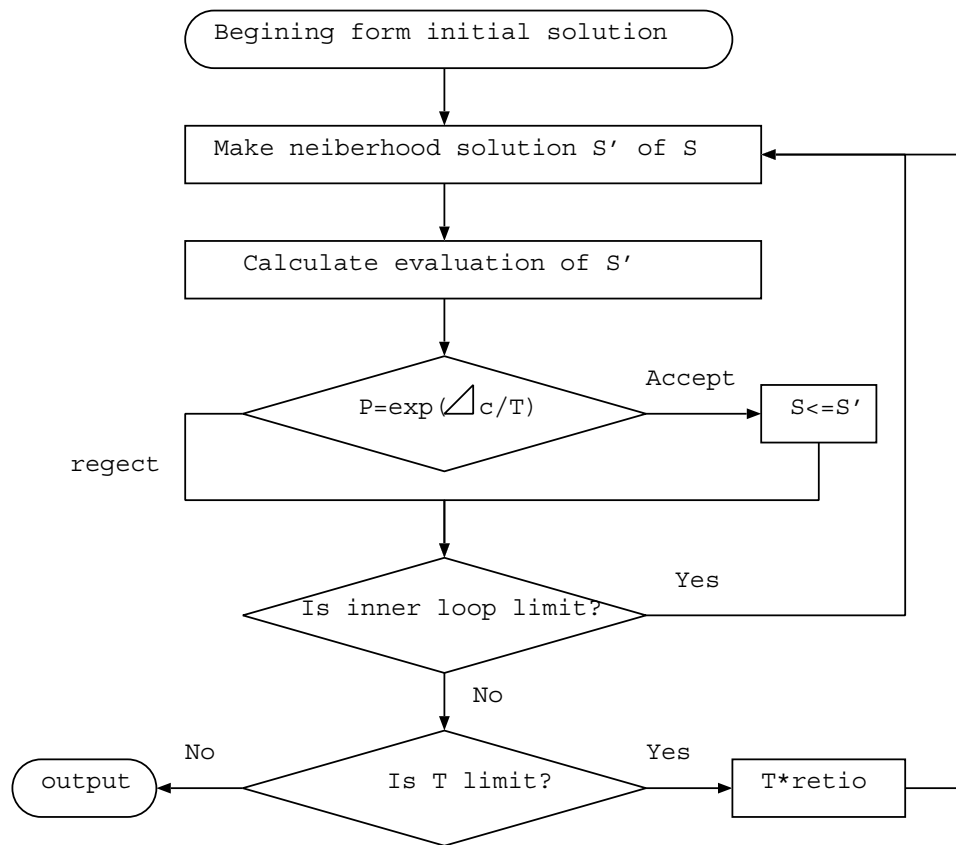


図 2.2: 一般的な SA のアルゴリズムの一例

第3章 提案手法

3.1 critical path

通常の矩形パッキングは、全ての矩形を囲む最小矩形 (Bounding-Box) の面積で解が評価される。このとき、Bounding-Box の幅と高さを決定するモジュールの並びが必ず存在し、このモジュールの並びを critical path と呼ぶ (図 3.1)。Bounding-Box 面積が縮小するためには、critical path が解消されなければならない。

SA では評価値の等しい隣接解は無定見に受理される。矩形パッキング問題では同じ評価値の隣接解が多数存在することから、解の変更が行なわれているにも関わらず一向に面積が良化しないという現象が見られる。これは、解の変更によって現存する critical path が解消されていないためである。例えば図 3.1 の配置は、水平方向では 1 回の隣接解生成操作では critical path を解消することができない (図 3.2)。しかしながら図 3.3 のように垂直方向では、1 回の隣接解生成操作で critical path を解消し面積を縮小できる。

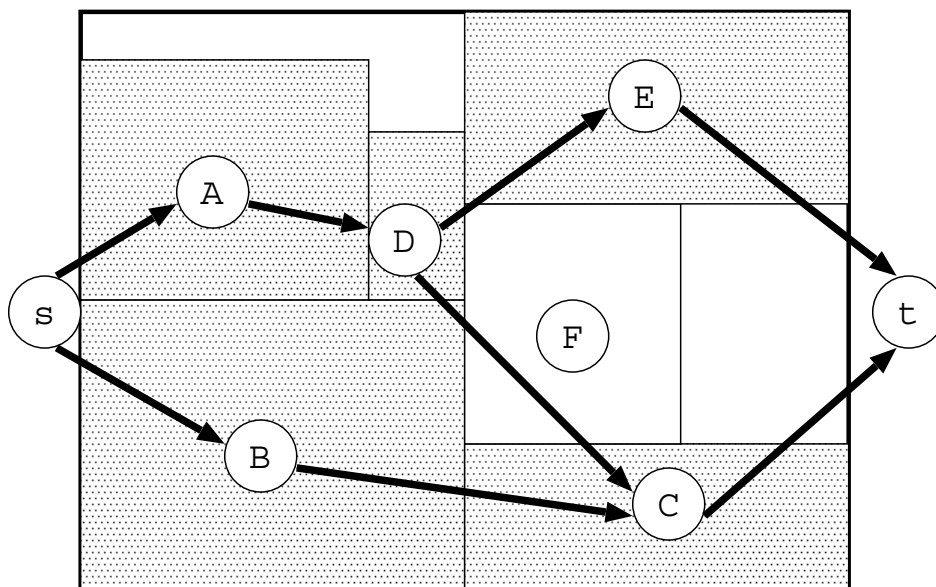


図 3.1: $\Gamma+(ADEBFC), \Gamma-(BADCFE)$ における水平 critical path

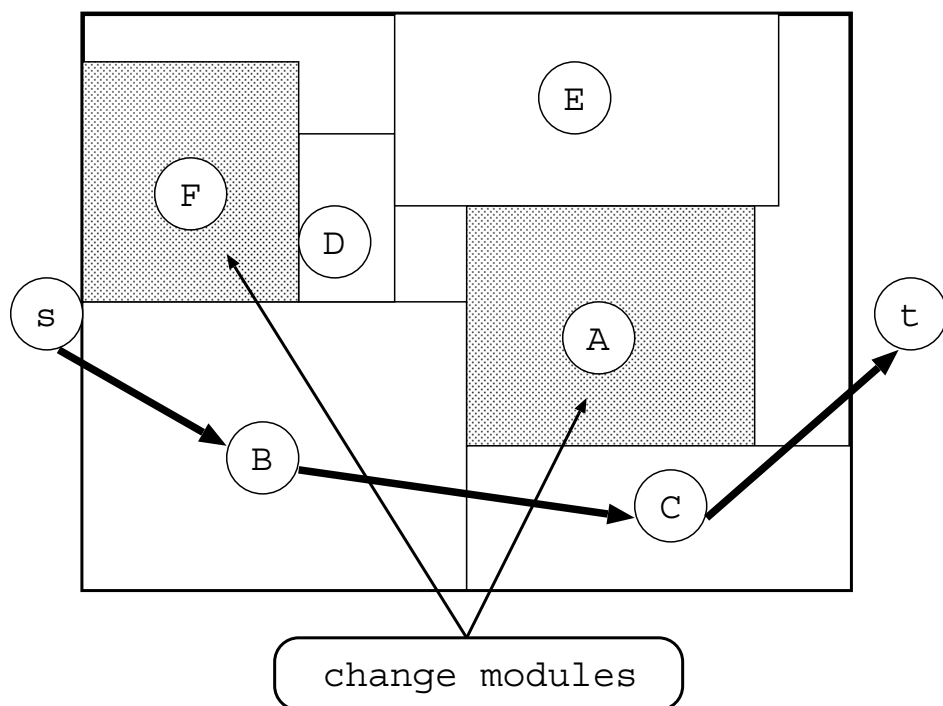


図 3.2: A と F の全交換 (水平 critical path が残っている)

3.2 カット次数

critical path の解消されやすさを評価するために、ここでカット次数と呼ぶ補助関数を導入する。カット次数は、与えられたレイアウトにおいて、現存する同一方向のすべての critical path が解消されるために変更の必要があるモジュールの最小個数である。これを水平方向、垂直方向ともに求める。図 3.1 の場合は横方向のカット次数が 2、図 3.2 では 1 であること、図 3.3 では縦方向のカット次数が 1 であることが分かる。

カット次数を導入することによって、面積縮小に至るまでに最低限必要な隣接解生成操作の繰り返し回数を評価することが可能となる。このカット次数の評価を SA に組み込み、同一面積の解の中でもよりカット次数の少ない解に探索を進めていくことで、探索効率を向上させることが可能である。

カット次数の算出は、次のように行なわれる。

1. Bounding-Box の左右端 (下上端) をそれぞれ source(s) と sink(t) とする。
2. 事前に、Sequence-Pair からパッキングを行なう時に、左端から右端 (下端から上端) までの longest path を求めておく。longest path を逆から辿ることによって、critical path 上にあるモジュールを抽出できるので、これを頂点とする。
3. 頂点と対応するモジュール間の相対位置関係を枝とするグラフ (critical path graph) を構成する。
4. critical path graph において、頂点の中を流すことのできるフローのキャパシティを

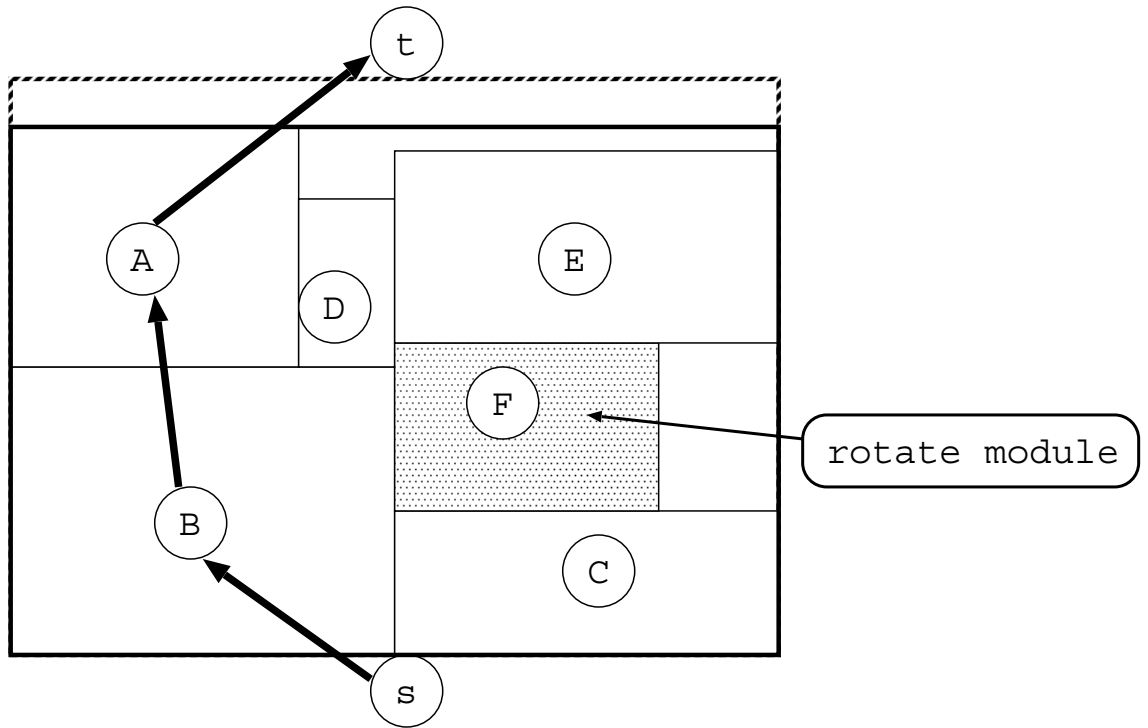


図 3.3: F の回転 (垂直方向 critical path が解消される)

- 1 とする、 s から t への最大フローを求める。
5. このとき、最大フロー-最小カット定理より、この最大フローの値がカット次数となる。

最大フローを求めるアルゴリズムの計算量は Ford と Fulkerson のアルゴリズムを用いるものとするれば、枝の容量の最大値 = U 、枝数 = m とした時、

$$O(m^2 * U)$$

である。ただし、critical path graph の枝容量はすべて 1 でよく、 t に入る枝は高々頂点の個数本であるので、1 回の反復で 1 ずつ流量が増加しても最大流量は頂点数を越えない。したがって最大反復回数の上限は高々頂点数回である。Ford と Fulkerson のアルゴリズムにおいて、各反復で 1 つの増加道を見つけ出すために必要な計算量は $O(m)$ であるので、頂点数を n とした時、

$$O(n * m)$$

で最大フロー、つまりカット次数を求めることができる。

カット次数を SA に組み込み評価することで、従来の SA では無定見に受理されていた、critical path が解消されやすい解から解消されにくい解への変更を抑制することができる。そして SA の内部ループ回数を削減すると、探索回数は減るので解の質は悪化するが、計算時間は短縮することができる。提案手法により探索を効率化すれば内部ループ回数を削

減しても解の質は保たれ、カット次数を計算する時間を差し引いても、従来の SA に比べて高速化が可能となる。

また、critical path 上のモジュール以外のモジュールを 2 個選んで変更を加えても、現存する critical path の解消にはつながらない。そこで隣接解生成操作で変更されるモジュールの中で、必ず 1 個は critical path 上のモジュールから選ばれるように改良を加えた。

3.3 カット次数の SA への組み込み

ここでは、面積が同じ解の中でもより critical path が解消されやすい解へと探索を進める方法として、SA で隣接解を受理するか否かの判定を面積、カット次数の 2 段階で行う方法と、面積とカット次数をまとめた 1 つの関数で 1 段階で行なう方法の 2 種類を提案する。

ある配置を表す Sequence-Pair のコードを S 、隣接解の Sequence-Pair のコードを S' とする。Bounding-Box の面積を $A(S)$ 、隣接解の Bounding-Box の面積を $A(S')$ とし、 $\Delta c_1 = A(S) - A(S')$ を面積の改善量とする。水平 (垂直) critical path graph の最小カット容量を $ch(S)(cv(S))$ 、隣接解の水平 (垂直) critical path graph の最小カット容量を $ch(S')(cv(S'))$ とする。縦、横どちらか一方の critical path を解消することができれば、Bounding-Box の面積は縮小する可能性があり、ここでは縦横のカット次数の最小値 $\min(ch(S), cv(S))$ を評価する方法と、縦横のカット次数の和 $ch(S) + cv(S)$ を評価する方法の 2 種類を提案する。

1. 面積、カット次数の順に 2 段階で判定する方法

- $A(S) > A(S')$ の場合は、 S' を受理し、 $S \leftarrow S'$ とする。次の隣接解との比較のために critical path graph のカット次数を計算する。
- $A(S) = A(S')$ あるいは $A(S) < A(S')$ で、 $\text{rand}(0, 1) \leq \exp(\Delta c_1 / T_1)$ で受理された場合、critical path graph のカット次数を計算し、 $\min(ch(S), cv(S)), \min(ch(S'), cv(S'))$ 、もしくは $(ch(S) + cv(S), ch(S') + cv(S'))$ を評価する第 2 段階に入る。
- $A(S) < A(S')$ で、 $\text{rand}(0, 1) > \exp(\Delta c_1 / T_1)$ で受理されない場合、 S' への変更を却下し、 S をそのまま保持する。

2. 面積、カット次数を組み合わせた関数で 1 段階で判定する方法

- Bounding-Box の面積、critical path graph のカット次数を計算し、2 つを組み合わせる ($Ac(S), Ac(S')$ とする)。
- $Ac(S) = Ac(S')$ あるいは $Ac(S) < Ac(S')$ で、 $\text{rand}(0, 1) \leq \exp(\Delta c / T)$ で受理された場合、 S' を受理し、 $S \leftarrow S'$ とする。
- $Ac(S) < Ac(S')$ で、 $\text{rand}(0, 1) > \exp(\Delta c / T)$ で受理されない場合、 S' への変更を却下し、 S をそのまま保持する。

上記の手法を C 言語を用いて計算機上に実装し、実験によって従来の SA との比較を行った。

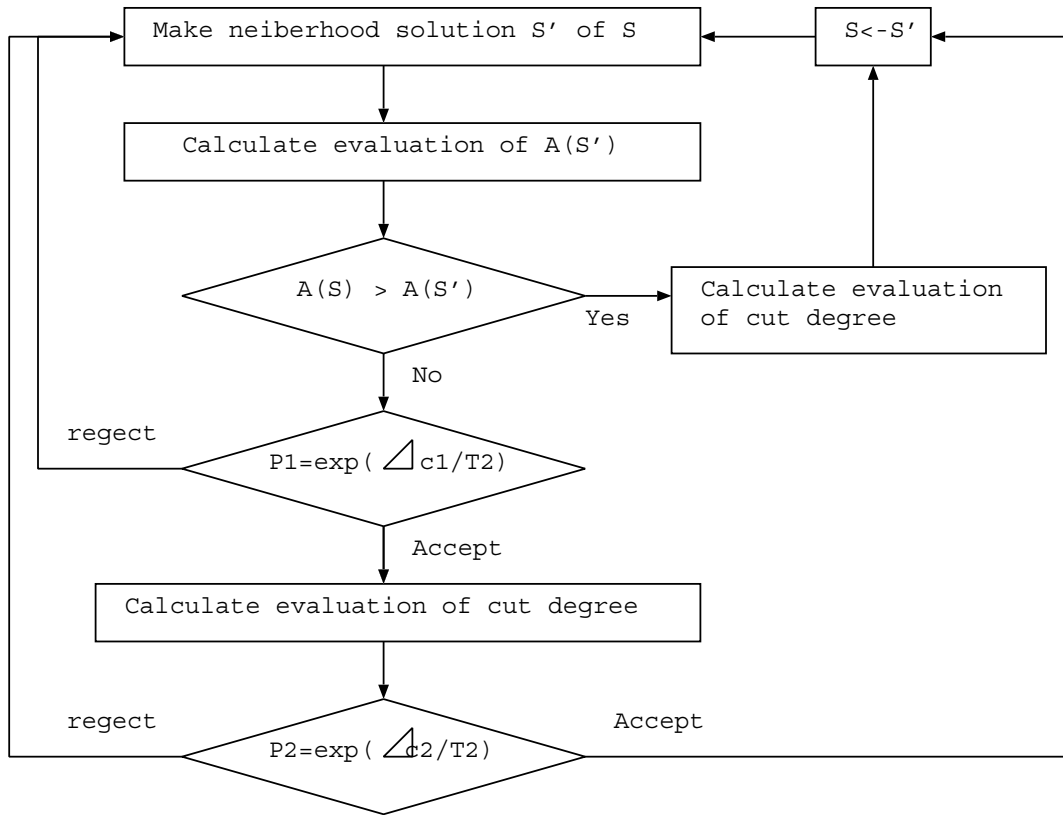


図 3.4: 面積、カット次数の順に 2 段階で評価を行なう SA のアルゴリズム

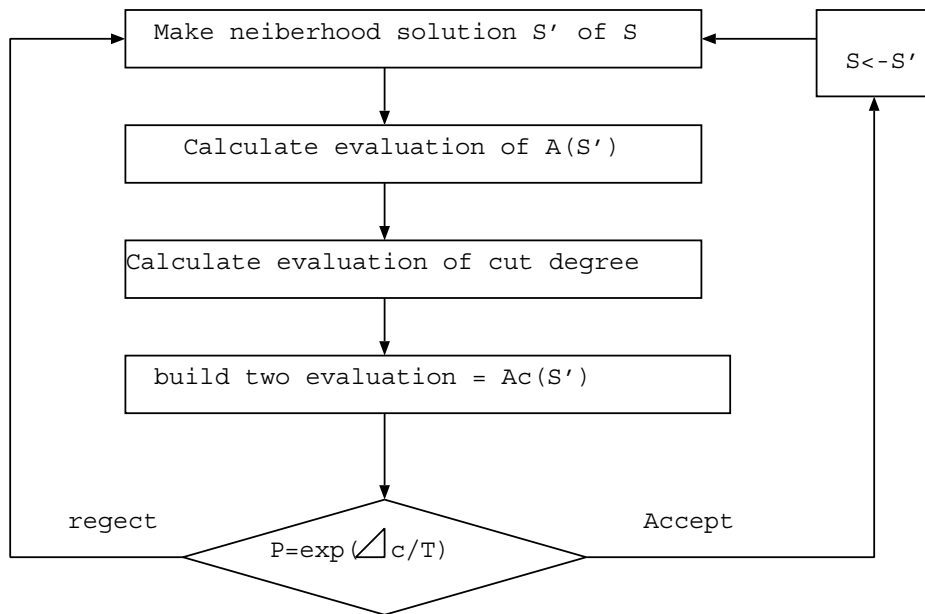


図 3.5: 面積、カット次数を組み合わせた評価関数で1段階で評価を行なう SA のアルゴリズム

第4章 実験と考察

4.1 準備

実験では1から10までの整数値の決められた幅、高さを持つ50個のモジュールを使用する。このモジュールセットを便宜上、m50と呼ぶことにする。さらに既存のベンチマークであるami49を用いた場合についても実験を行なう。

Bounding-Boxの面積を A 、モジュールの総面積を B とし、面積率を以下のように定義する。

$$((A - B)/(B)) \times 100 \quad (4.1)$$

面積率は隙間の全くない理想的な配置と比べてどの程度 Bounding-Box の面積が悪化しているかを表し、実験ではSA終了時の面積率を比較するものとする。

SAで使用する隣接解は、以下の3種類の中から等確率で選択する。

- 全交換：選択された2つのモジュールについて、 Γ_+ 、 Γ_- の両方で入れ替える。
- Γ_+ 交換：選択された2つのモジュールについて、 Γ_+ のみ入れ替える。
- 回転：選択された1つのモジュールを90度回転する。

SAの開始温度は、すべてのモジュール面積の総和を最良、すべてのモジュールの幅の総和とすべてのモジュールの高さの総和を掛けたものを最悪とし、最良から最悪に遷移した場合でも約50%の確率で受理される100000度とした。また、実験を繰り返した結果、m50では1度から0.01度の間ではほぼ解の向上が見込めなくなることから、0.1度を終了温度として設定した。

温度降下係数は0.98、内部ループは1000回とした。実験結果はそれぞれ異なる初期解から開始した10回の統計をとったものである。

4.2 実験

4.2.1 実験1：カット次数を評価する2段階目の温度パラメータに関する実験と比較

カット次数を用いて2段階目の判定を行う際の温度スケジュールについて、実験によって妥当な開始温度を求める。ここでは水平、垂直のカット次数の最小値を評価、2段階目の温度パラメータを T_2 とし、 T_2 の開始温度をそれぞれ 100、10、1 として実験を行った。なお T_2 で使用する降下係数は、SA 終了時に T_2 の温度が 0.01 前後になるように調整している。実験結果を表 4.1 に示す。

4.2.2 実験2：従来のSAと、カット次数を用いたSAとの比較

ここではカット次数を用いて2段階で受理不受理の判定を行なう方法と、従来の面積のみを評価するSAとの比較、さらに critical path 上のモジュールが必ず1つは変更モジュールとして選択されるようにした場合と、ランダムにモジュールを選択した場合との比較を行なった。実験結果を表 4.2 に示す。

4.2.3 実験3：縦横のカット次数の最小値を評価した時と、和を評価した時の比較

ここではカット次数を用いて2段階で判定を行なう方法で、評価する値を縦横のカット次数の最小値とした時と、縦横のカット次数の和とした時の比較を行なった。実験結果を表 4.3 に示す。

4.2.4 実験4：面積、カット次数の2段階で評価した時と、面積とカット次数を組み合わせた関数を1段階で評価した時の比較

ここでは面積、カット次数の2段階で評価を行なう方法と、面積とカット次数を組み合わせた1つの評価関数を用意して1段階で評価する方法の比較を行なった。評価関数は、以下の2種類を使用した。

- 面積 + カット次数
- 面積 $\times 10$ + カット次数

実験結果を表 4.4 に示す。面積 $\times 10$ + カット次数を評価関数とした場合は、評価関数の値がほぼ 10 倍になるので、SA の開始温度、終了温度をそれぞれ 10 倍の 1000000 度、1 度にして調整をしている。

4.2.5 実験5：カット次数、モジュール選択ありでSAの高速化を行なった場合と、従来のSAとの比較

ここでは面積、カット次数の2段階で評価を行なう提案手法の内部ループの回数を、1000回から100回にすることで探索時間を1/10に短縮し、その結果について従来のSAとの比較を行なった。さらに従来SAも内部ループの回数を100回、500回、1000回と変化させ、結果について考察をした。実験結果を表4.5に示す。

4.2.6 実験6：ベンチマークami49を用いた場合での比較

最後にMCNCベンチマークami49に適用した場合で、提案手法と従来のSAとの比較を行なった。縦横のカット次数の最小値を2段階で評価し、 T_2 の開始温度を100度、10度とした場合と、従来手法でcritical path上のモジュールが必ず1つは変更モジュールとして選択されるようにした場合、まったくランダムにモジュールを選択した場合との比較を行なった。実験結果を表4.6に示す。なお、ami49は各モジュールの面積がm50に比べて非常に大きいので、開始温度を20億度、終了温度を2000度としている。

4.3 実験結果の考察

4.3.1 実験1：カット次数を評価する2段階目の温度パラメータに関する考察

表4.1より、 T_2 開始温度を10度とした場合が平均的に良い結果を出している。図4.2からも分かるように、 T_2 開始温度1の時は比較的高い温度で収束しており、2段階目の温度パラメータがSAの収束に影響を与えていることが分かる。

実験より、 T_2 開始温度100ではカット次数の効果がやや薄く、 T_2 開始温度1では局所最適に陥る可能性があるものと思われる。よって、これ以降の実験結果は基本的に T_2 開始温度10度の場合を用いて比較する。

4.3.2 実験2：従来のSAと、カット次数を用いたSAにおける実験結果の考察

表4.2、図4.3より、カット次数、モジュール選択ともになしの従来SAと、カット次数、モジュール選択共にありの提案手法を比較すると、計算時間は約1.5倍に悪化するものの平均、最良、最悪いずれも良い結果を出しており、提案手法が有効であることが確認できた。温度100から0.1までを拡大した図4.4より、カット次数を導入した場合は低温域でも良解を見つけることができることが分かる。これはカット次数が増える隣接解を不受理に

することにより、探索効率が向上しているためである。各温度における受理回数(図 4.7)を見ると、カット次数を用いた提案手法では従来手法に比べて受理回数が減っており、面積が同じ配置の中でカット次数が増える隣接解を不受理にしていることが確認できる。

さらに図 4.6 より、変更モジュールとして critical path 上のモジュールを必ず一つ選択するように改良した場合は、探索開始直後から選択しない場合と比べてやや低い面積率になることが分かる。これは探索開始直後の面積率が高い状態ではほとんどの場合カット次数は縦横とも 1 であり、critical path 上のモジュールを変更することで現在ある critical path が解消されて、面積が縮小する隣接解を多く生成するためと考えられる。

モジュール選択の有無だけでは最終結果にそれほど違いが見られない原因は、探索が進むにつれてほとんどのモジュールが critical path 上のモジュールになるので、ランダムで選択した場合と差がなくなるためである。

なお、critical path 上のモジュールを確認するためにパスを逆向きに辿る操作の計算時間が余分にかかっているため、カット次数なし、モジュール選択ありの場合と、カット次数、モジュール選択ともになしの場合の計算時間に差が生じている。

4.3.3 実験 3 : 縦横のカット次数の最小値を評価した時と、和を評価した時の比較に関する考察

表 4.3 より、縦横のカット次数の最小値を評価した場合に良い結果を出していることが分かる。図 4.8 のように、縦横のカット次数の和を評価した場合、最小値を評価した場合と比べて若干早めに収束に向かうことから、面積が縮小するにつれて縦横のカット次数の和の増加を許さないという制限が厳しく働き局所最適に陥る可能性が高いためと推測される。

4.3.4 実験 4 : 面積、カット次数の 2 段階で評価した時と、面積とカット次数を組み合わせた関数を 1 段階で評価した時の比較に関する考察

表 4.4 より、面積、カット次数の 2 段階で評価した場合が良い結果を出している。面積 + カット次数、面積 $\times 10$ + カット次数の 2 種類の評価関数は、面積が縮小してカット次数が増大した時に評価値の悪化があり得る場合と、面積が縮小してカット次数が増大した時に評価値の悪化がない場合として採用した。

この実験では、面積とカット次数を組み合わせた評価関数の作り方についての考察が不十分であるため良い結果を得られなかった。また、組み合わせる手法では必ずカット次数を計算するのに対し、2 段階で評価する手法では面積が拡大して不受理となった場合にカット次数を計算する必要がないという利点があり、これが計算時間の差となって現れている。

表 4.1: 実験 1 の結果 : カット次数を評価する 2 段階目の温度パラメーターに関する実験と比較

T2 開始温度	平均面積率 (%)	最良	最悪	計算時間
100	4.1200	2.4949	6.2036	41.1s
10	3.7896	2.2927	4.7876	41.1s
1	4.2819	1.6858	6.8105	41.1s

表 4.2: 実験 2 の結果 : 従来の SA と、カット次数を用いた SA との比較

	平均面積率 (%)	最良	最悪	計算時間
カット次数、モジュール選択ともにあり	3.7896	2.2927	4.7876	41.1s
カット次数あり、モジュール選択なし	4.0931	1.9555	5.1922	41.9s
カット次数なし、モジュール選択あり	6.8173	5.0573	7.8894	33.9s
カット次数、モジュール選択ともになし	7.0263	5.0573	9.2380	26.6s

4.3.5 実験 5 : カット次数、モジュール選択ありで SA の高速化を行なった場合と、従来の SA との比較に関する考察

表 4.5 より、カット次数、モジュール選択ありで内部ループ 100 回にした場合でも、すべてなしで内部ループ 1000 回の場合と比較して良い結果を出している。計算時間は従来の SA と比べて約 1/6 なので、提案手法により探索の高速化が実現している。また、すべてなしの時では内部ループ 100 回、500 回では面積率がやや悪化してしまうため、内部ループ 100 回、500 回では試行回数がまだ不十分であると言える。

4.3.6 実験 6 : ベンチマーク ami49 を用いた場合での実験結果に関する考察

表 4.6 より、ここでは提案手法と従来の SA との差が見られなかった。ami49 の様に各モジュールの幅、高さのばらつきが大きいと、モジュールの並びが探索が進んだ低温状態であってもカット次数は縦、横それぞれ 1 の場合がほとんどである。提案手法と従来手法に差が見られないのは、カット次数の増減がほとんどないためにカット次数を評価する提案手法の利点を生かすことができないためと考えられる。

表 4.3: 実験 3 の結果 : 縦横のカット次数の最小値を評価した時と、和を評価した時の比較

	平均面積率 (%)	最良	最悪	計算時間
縦横のカット次数の最小値を評価	3.7896	2.2927	4.7876	41.1s
縦横のカット次数の和を評価	4.6864	3.5738	5.7316	41.1s

表 4.4: 実験 4 の結果 : 面積、カット次数の 2 段階で評価した時と、面積とカット次数を組み合わせた関数を 1 段階で評価した時の比較

	平均面積率 (%)	最良	最悪	計算時間
面積、カット次数の 2 段階で評価	3.7896	2.2927	4.7876	41.1s
(面積 + カット次数) を 1 段階で評価	4.7269	3.1693	6.2036	52.3s
(面積 * 10 + カット次数) を 1 段階で評価	6.4127	4.383	7.2825	52.3s

表 4.5: 実験 5 の結果 : カット次数、モジュール選択を用いて SA の高速化を行なった場合と、従来の SA との比較

	平均面積率 (%)	最良	最悪	計算時間
両方ともありで内部ループ 100 回	6.8712	4.383	8.5637	4.1s
すべてなしで内部ループ 100 回	10.2832	7.2825	13.284	2.6s
すべてなしで内部ループ 500 回	7.9434	6.6082	10.182	13.3s
すべてなしで内部ループ 1000 回	7.0263	5.0573	9.2380	26.6s

表 4.6: 実験 6 の結果 : ベンチマーク ami49 を用いた場合での比較

	T_2 開始温度	平均面積率 (%)	最良	最悪	計算時間
両方ともあり	100	3.8127	2.9207	6.5559	41.2s
両方ともあり	10	3.8332	3.1430	4.5155	41.9s
モジュール選択のみあり		3.9393	3.2558	5.3073	33.98s
すべてなし		3.8230	2.9462	5.5551	26.6s

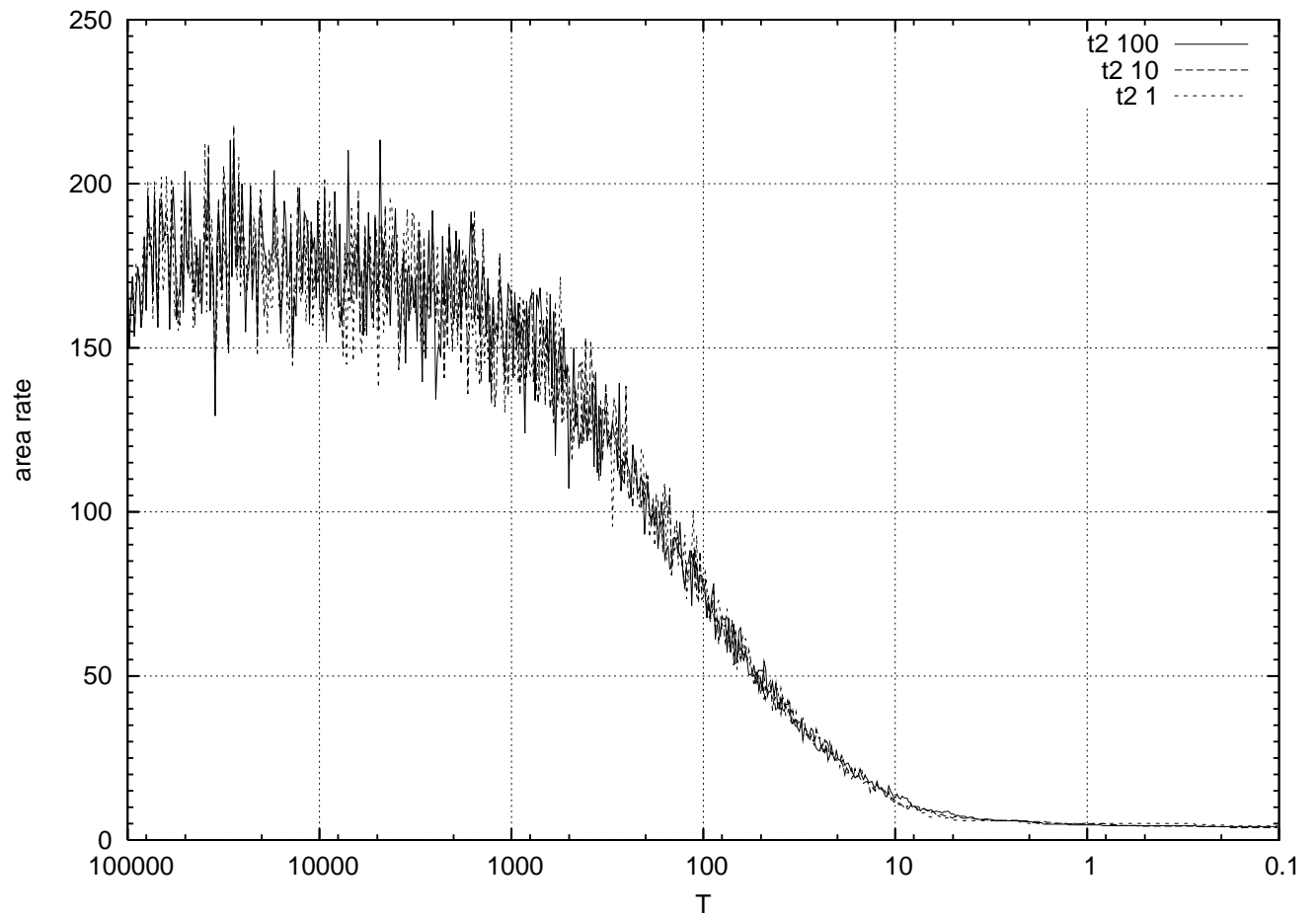


図 4.1: 実験 1 の結果 : カット次数を評価する 2 段階目の温度パラメーターに関する実験と比較

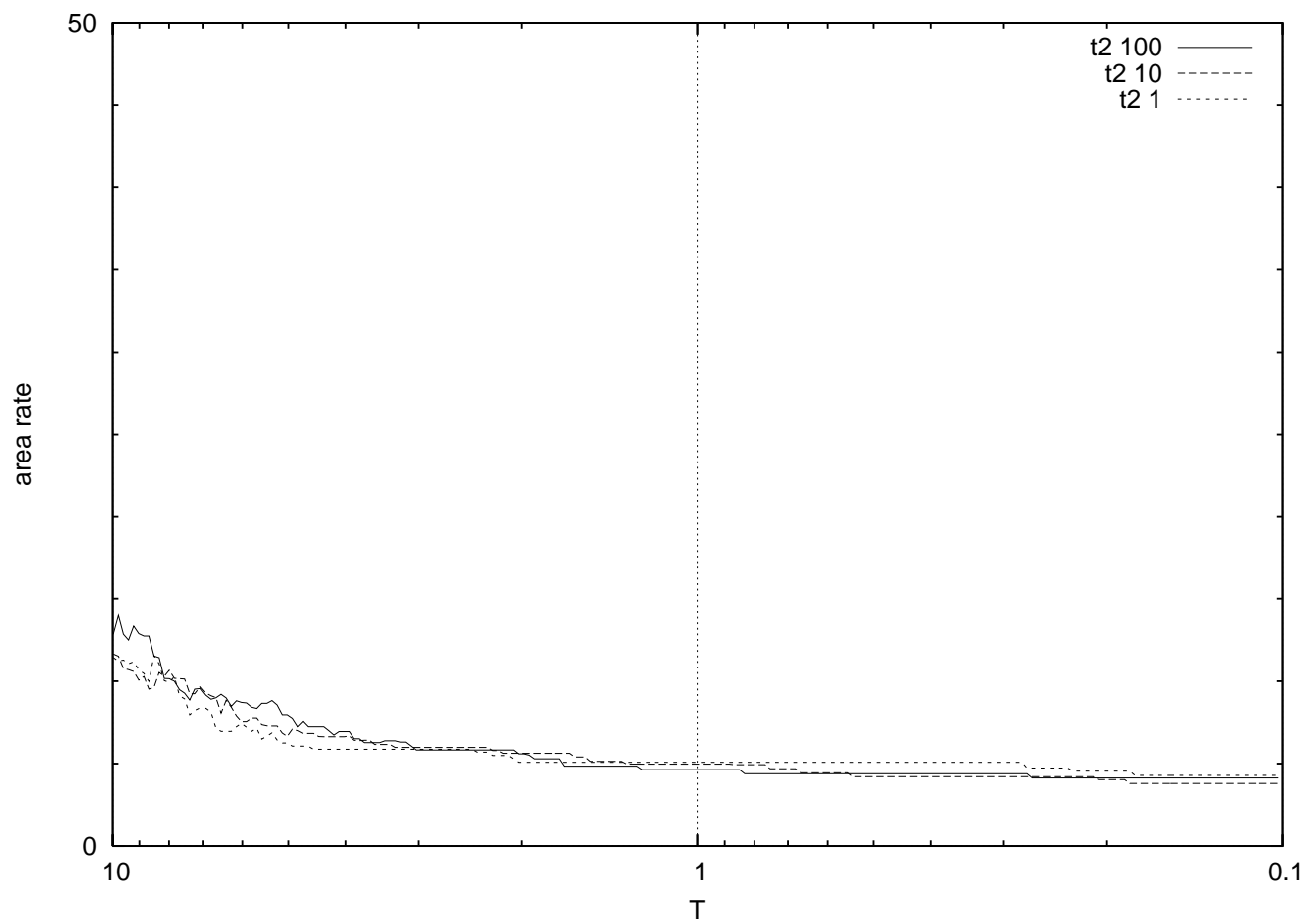


図 4.2: 実験 1 の結果 : 温度 10 から 0.1 までを拡大

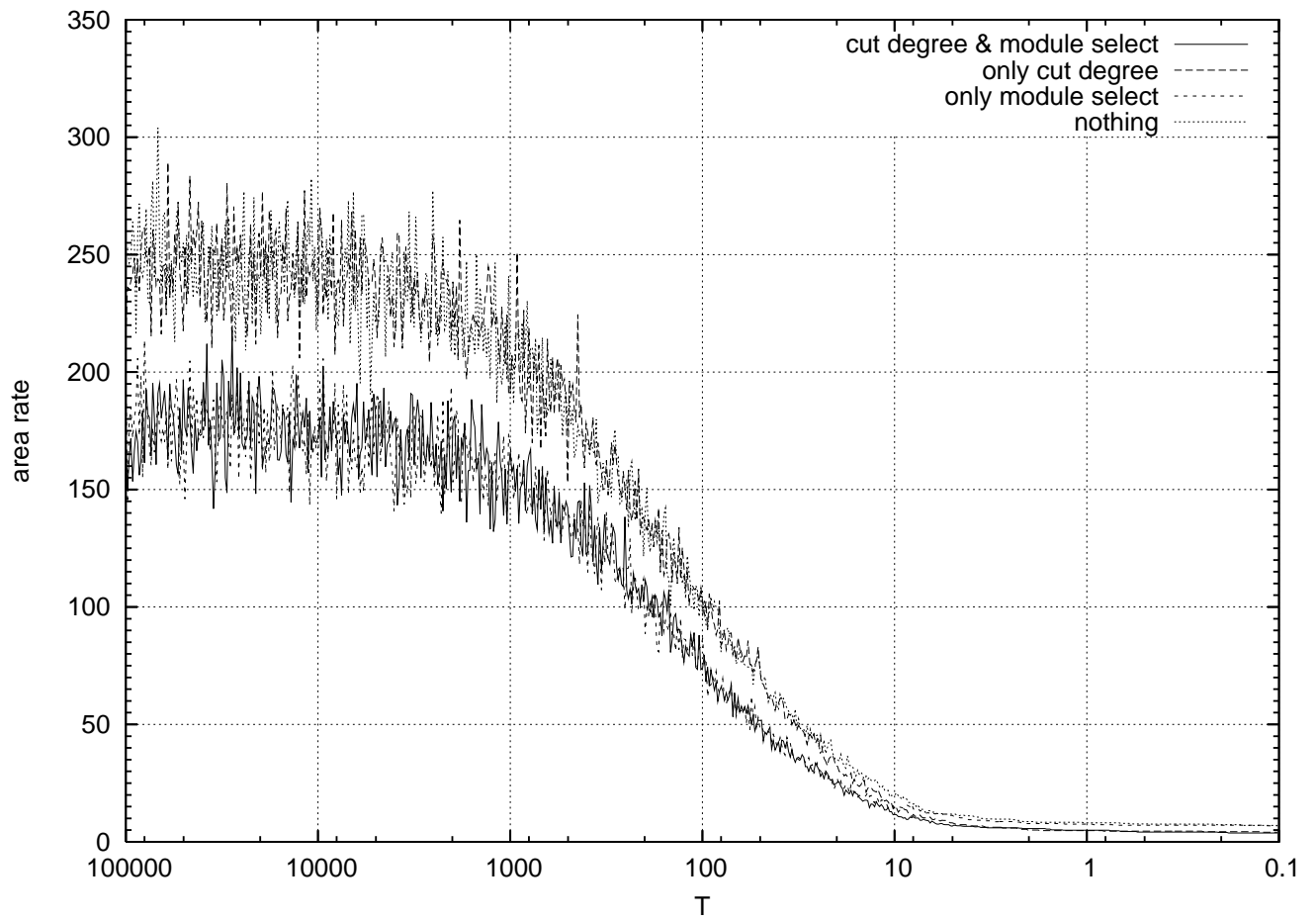


図 4.3: 実験 2 の結果 : すべてあり、カット次数のみ、モジュール選択のみ、すべてありの 4 系統

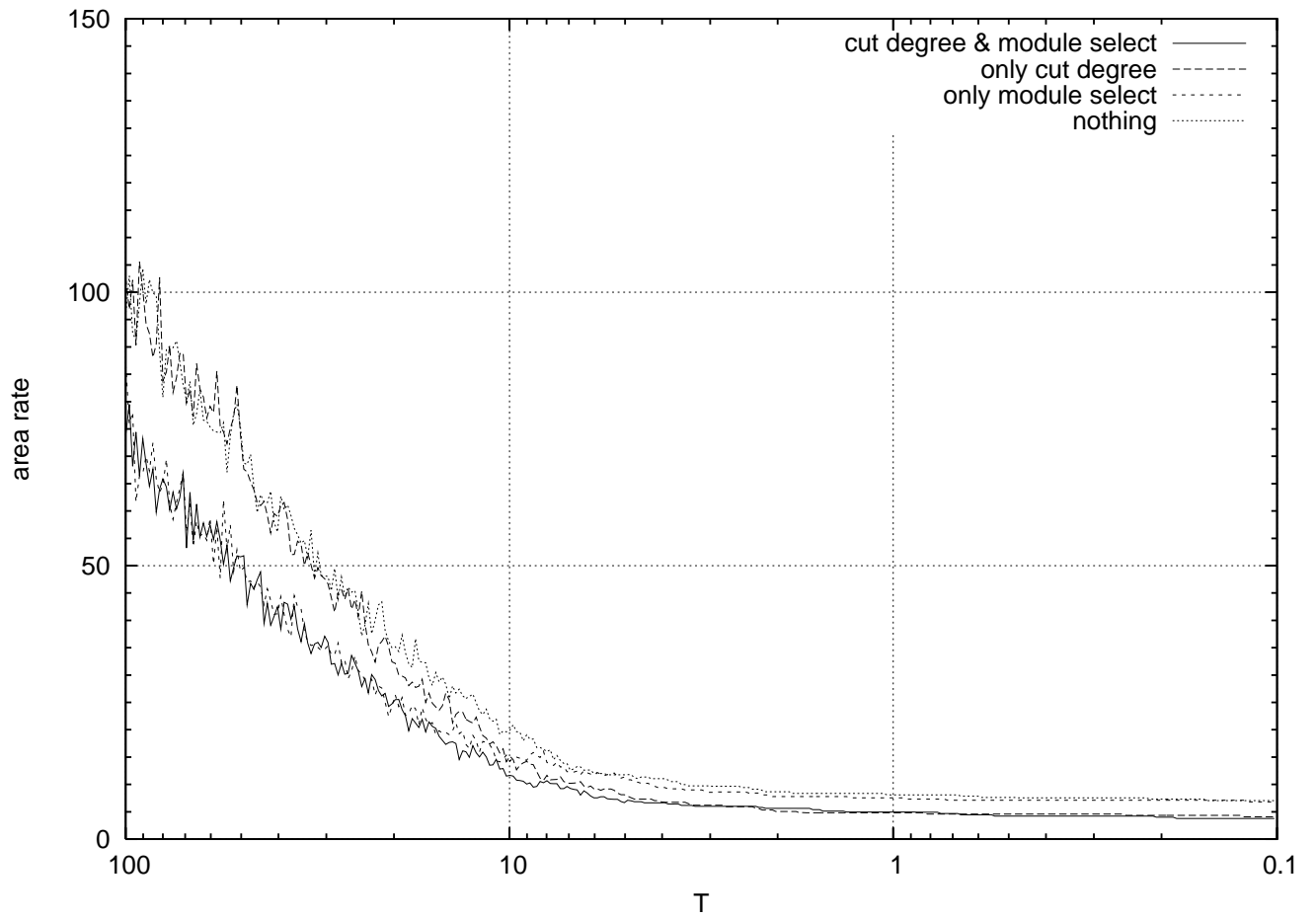


図 4.4: 実験 2 の結果 : 温度 100 から 0.1 までを拡大

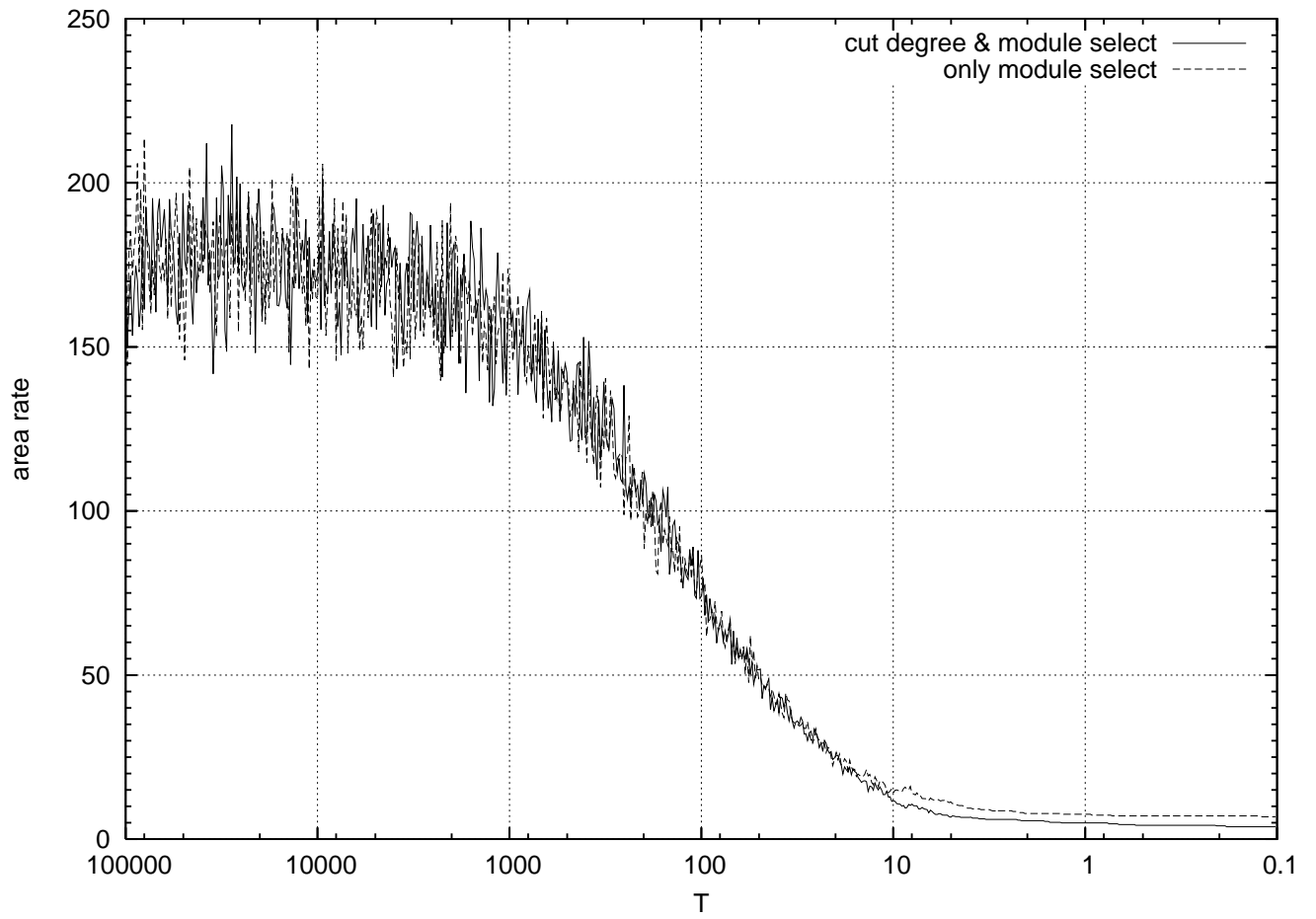


図 4.5: 実験 2 の結果 : すべてありと、モジュール選択のみの 2 系統

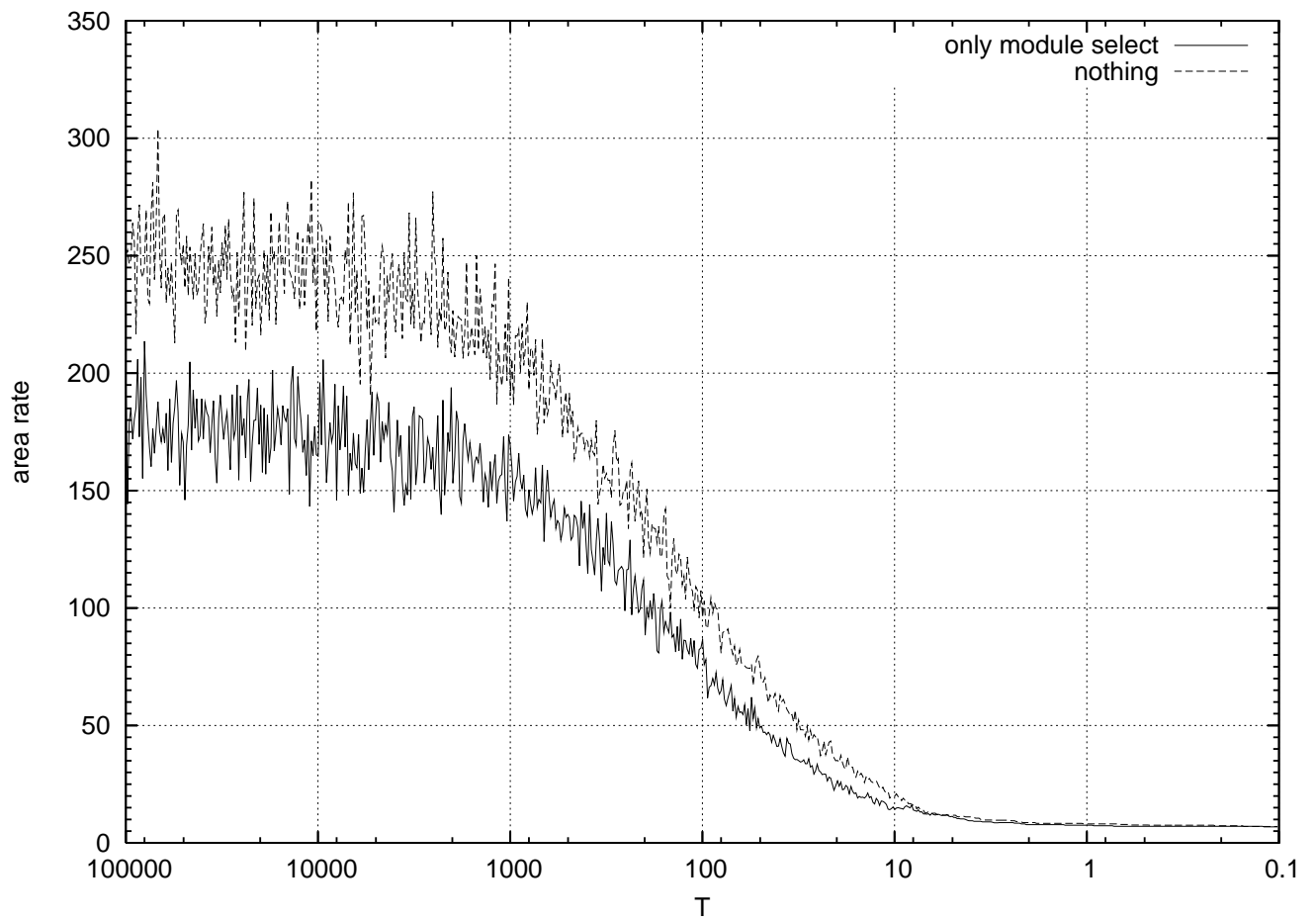


図 4.6: 実験 2 の結果 : モジュール選択のみと、すべてなしの 2 系統

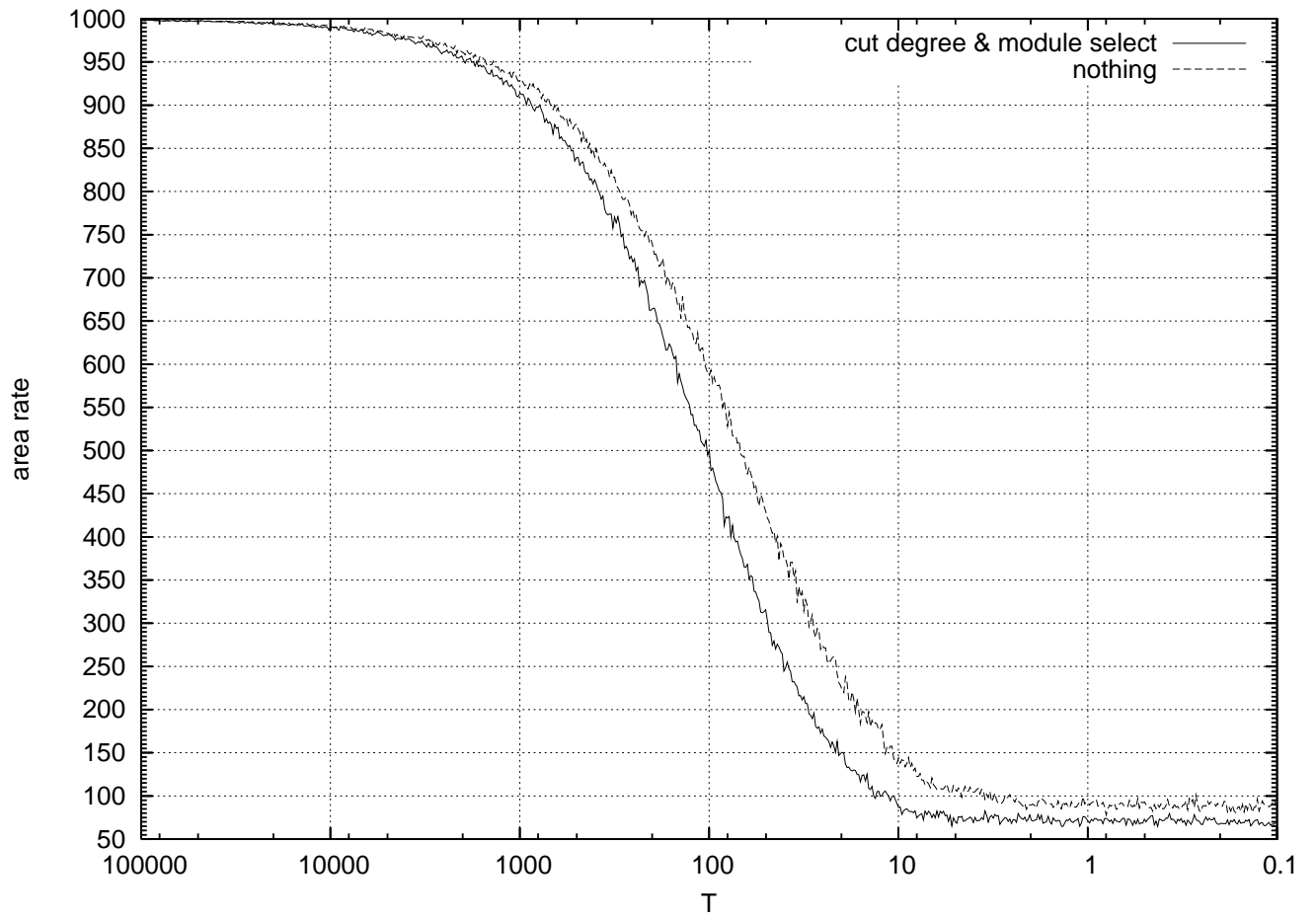


図 4.7: 実験 2 における受理回数の推移 : すべてありと、すべてなしの 2 系統

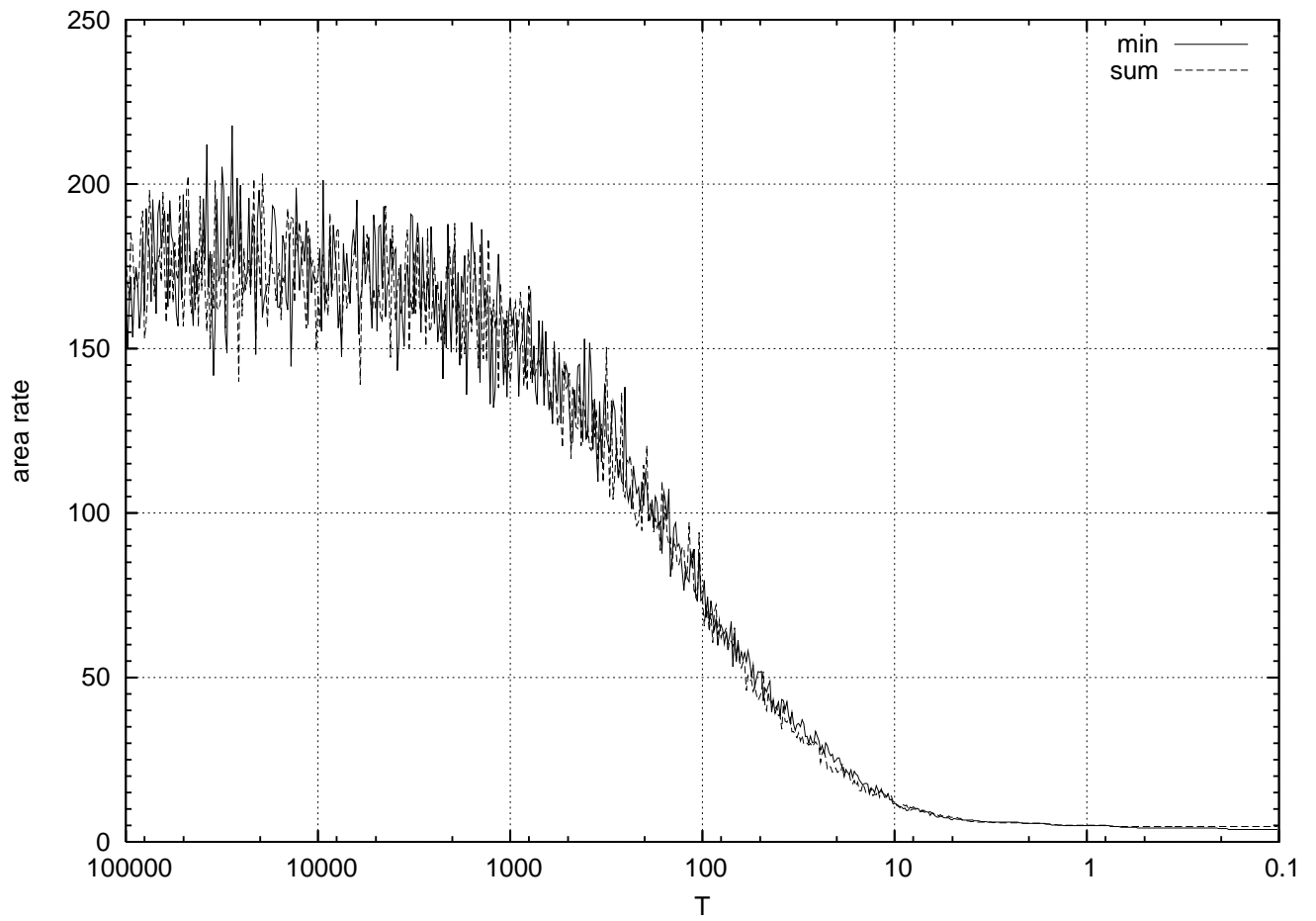


図 4.8: 実験 3 の結果 : 縦横のカット次数の最小値を評価した時と、和を評価した時の 2 系統

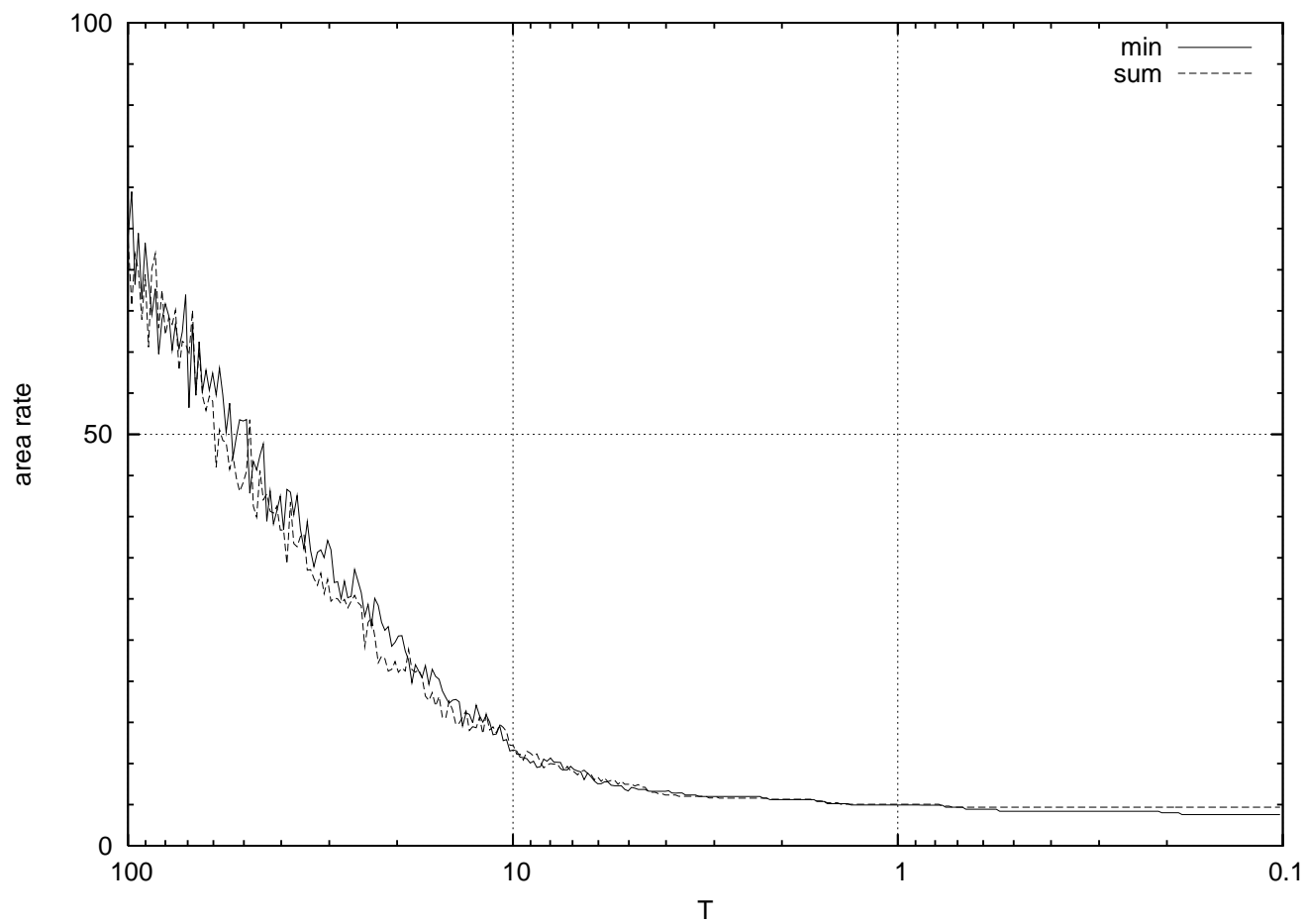


図 4.9: 実験 3 の結果 : 温度 100 から 1 までを拡大

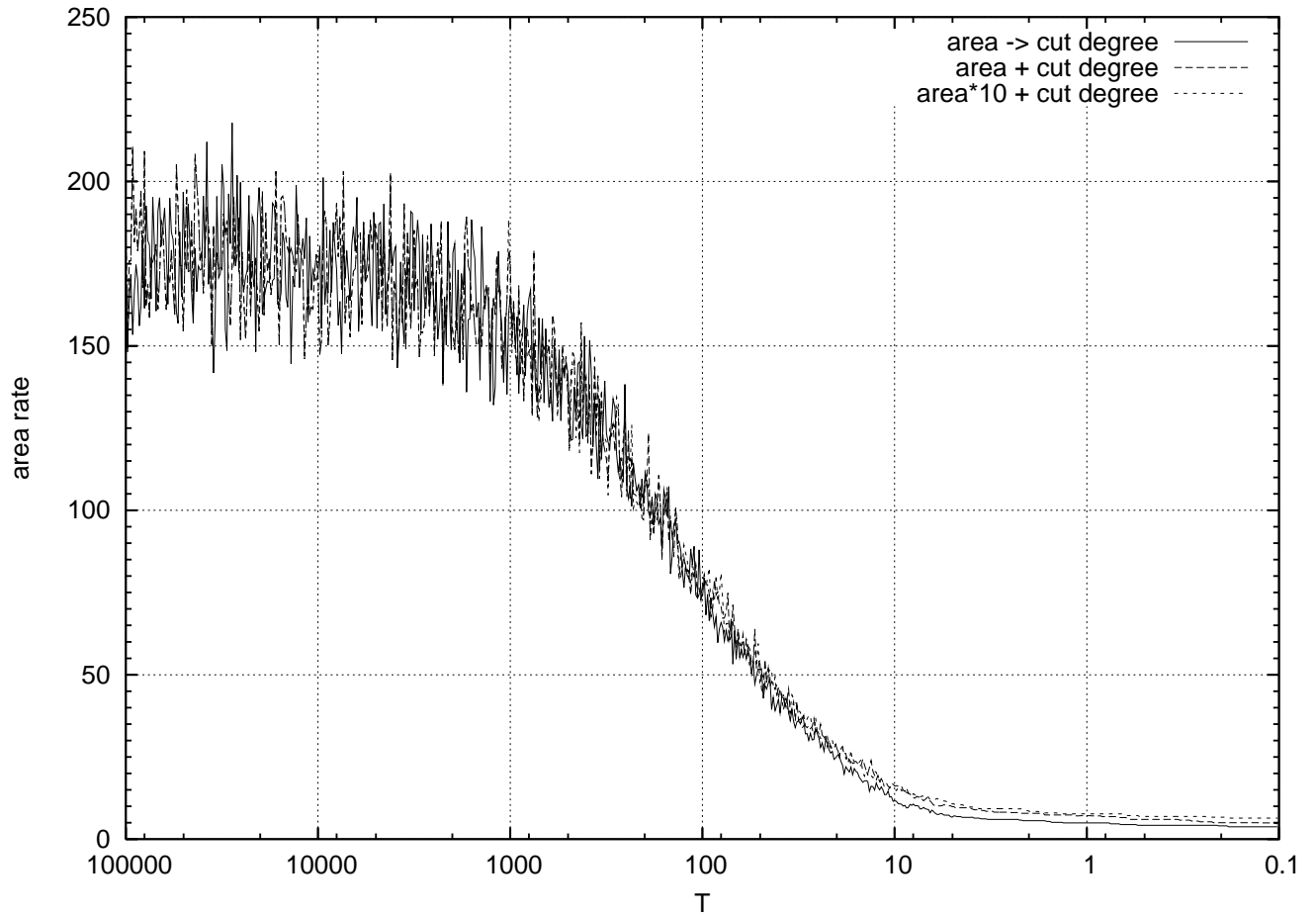


図 4.10: 実験 4 の結果：面積、カット次数の 2 段階で評価した時と、面積とカット次数を組み合わせ関数を 1 段階で評価した時 (面積 + カット次数、面積 $\times 10$ + カット次数) の 3 系統 (計算時間を合わせるため、面積 $\times 10$ + カット次数では T は 1000000 から 1 まで)

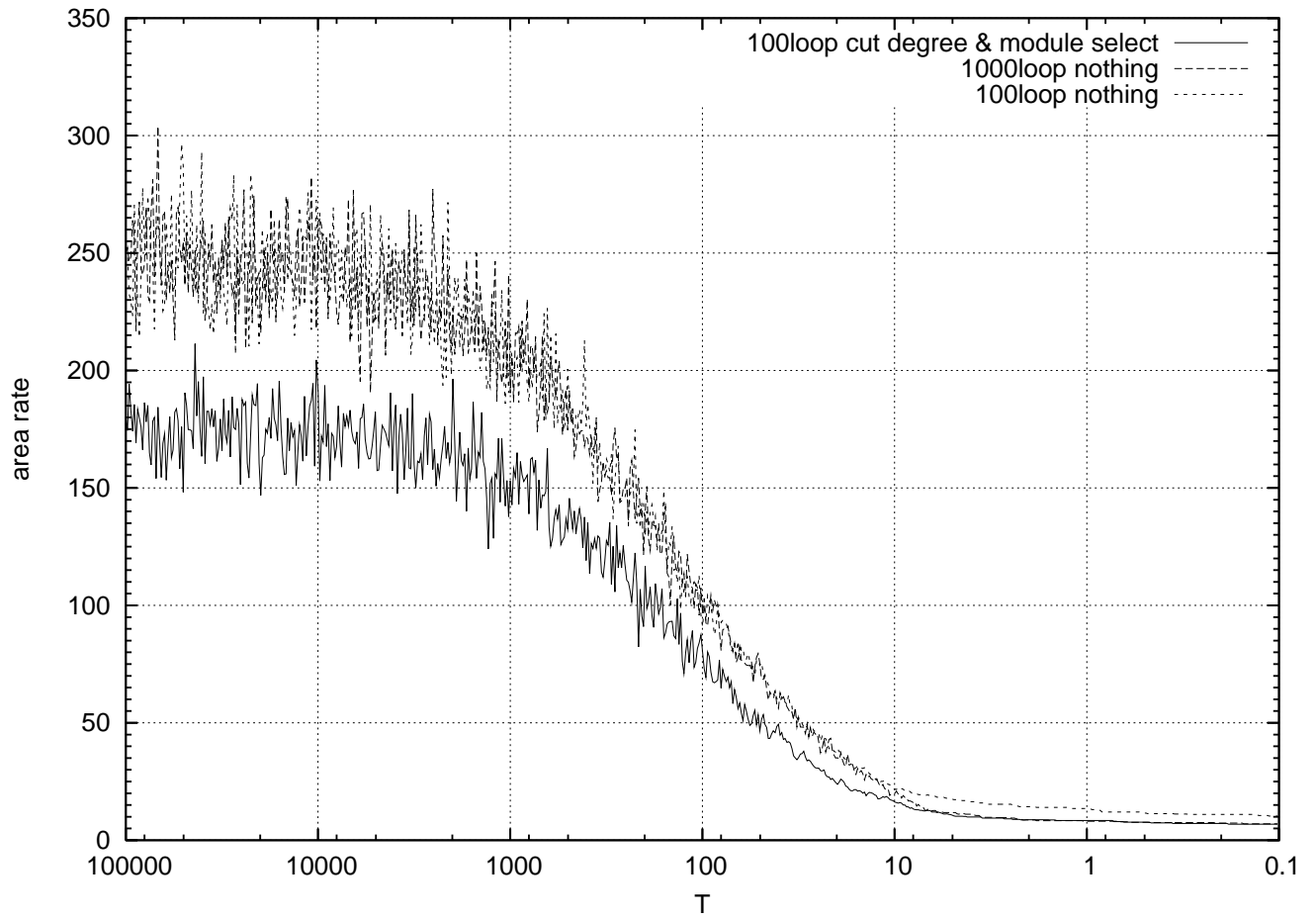


図 4.11: 実験 5 の結果 : カット次数、モジュール選択を用いて SA の高速化を行なった場合 (内部ループ 100 回) と、従来 SA (内部ループ 1000 回と 100 回) の 3 系統

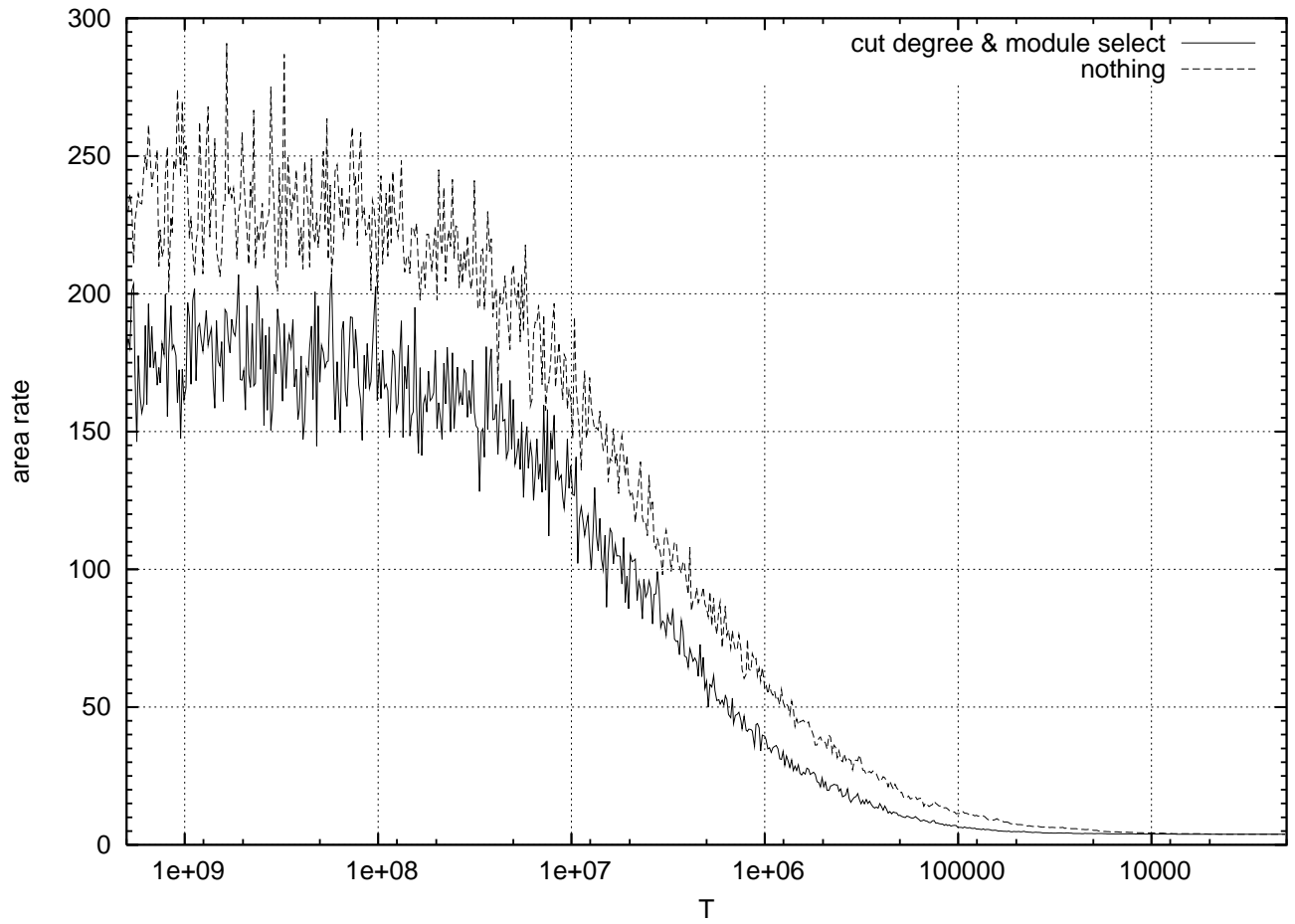


図 4.12: 実験 6 の結果 : ベンチマーク ami49 を用いた場合での比較 (すべてありと、すべてなしの 2 系統)

第5章 結論

5.1 まとめ

本研究では矩形パッキング問題に対して、Bounding-Box の幅、高さを決定している critical path に注目した。critical path の解消されやすさを評価する、カット次数と呼ぶ補助関数を導入し、これを SA に組み込むことで解探索の効率化を目指した。

実験の結果、それぞれのモジュールの幅、高さのばらつきが少ないモジュールセットでは提案手法が有効に働くことが確認できた。一方、ami49 のようにモジュールの幅、高さのばらつきが大きいモジュールセットでは提案手法の有効性が認められなかった。

5.2 今後の課題

提案手法が有効でなかった ami49 のように、モジュールの幅、高さのばらつきが大きいモジュールセットでも有効に働く、新たな補助関数を導入する必要がある。また本研究では、Bounding-Box の面積だけを評価する矩形パッキング問題を中心に考えたが、実際の VLSI チップレイアウト設計では配線長も最適化の対象となる。面積の最小化と配線長の最小化を同時に考える、実際のレイアウト設計における補助目的関数の導入と SA の効率化は、今後の課題である。

第6章 謝辞

本研究を進めるにあたり、多大なる御指導を受け賜りました金子峰雄教授、ならびに多くの協力を頂きました研究室の皆様に、深く感謝と御礼を申し上げます。

参考文献

- [1] Hiroshi Murata, Kunihiro Fujiyoshi, Shigetoshi Nakatake, Yoji Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL.15, NO.12 pp.1518-1524,(1996-12)
- [2] 藤吉 邦洋, 大村 智一, 井尻 堅大, "Simulated Annealing 法探索に適した Sequence-Pair によるパッキング解空間", TECHNICAL REPORT OF IEICE VLD99-118 ICD99-275 pp.9-16,(2000-03).
- [3] Satoshi Tayu, Takayuki Obata, Mineo Kaneko, "Efficient Search on Solition Space Based on Sequence-Pair for Simulated Annealing Approach" TECHNICAL REPORT OF IEICE VLD2002-5 pp.25-30,(2002-05).
- [4] 内田 誠司, 高橋 篤司, "Move の制限によるシミュレテッドアニーリング法を用いたパッキングの高速化", TECHNICAL REPORT OF IEICE CAS2002-17 VLD2002-31 DSP2002-57 pp.95-100,(2002-06).
- [5] 内田 誠司, 高橋 篤司, "低温域における Move の制限による Simulated Annealing 法を用いたパッキングの高速化", IPSJ SIG Technical Report 2003-SLDM-112 pp.193-198,(2003-11).
- [6] 今堀 慎治, 柳浦 睦憲, 茨木 俊秀, "配置コストをもつ長方形詰込み問題に対する局所探索法の高速化", 第 87 回アルゴリズム研究会 pp.51-58,(2002-11).
- [7] 澤 卓, 千原 國宏, 嵩 忠雄, 佐藤 宏介, "方形パッキング手法による MMIC 向き配置配線の自動化に関する一手法", 奈良先端科学技術大学院大学 修士論文 NAIST-IS-MT9651054, (1998-2).