

<b>Title</b>	均一でない計算機を利用した並列分枝限定法に関する考察
<b>Author</b>	大西, 克実 / 中野, 秀男
<b>Citation</b>	大阪市立大学学術情報総合センター紀要. Vol. 4, p.59-64.
<b>Issue Date</b>	2003-03
<b>ISSN</b>	1345-4145
<b>Type</b>	Departmental Bulletin Paper
<b>Textversion</b>	Publisher
<b>Publisher</b>	大阪市立大学学術情報総合センター
<b>Description</b>	

Placed on: 大阪市立大学学術機関リポジトリ

Placed on: Osaka City University Repository

## 均一でない計算機を利用した並列分枝限定法に関する考察

大西克実, 中野秀男<sup>†</sup>

最近の PC およびネットワーク技術の発達により従来の並列コンピュータとは異なったメタコンピューティング環境上での並列分枝限定法の研究が行われている。並列分枝限定法は、組合せ最適化問題の最適解を求めるために利用される分枝限定法を並列化した解法である。メタコンピューティング環境では従来の並列処理環境とは異なり計算機環境を構成する各計算機の能力は必ずしも均一ではない。本報告では性能の異なる 2 種類の計算機から構成される環境上で並列分枝限定法を実現し、異なる計算機を利用する影響を単一の計算機で構成した時の結果と比較検討する。

キーワード 組合せ最適化問題, 分枝限定法, 並列処理

### Parallel branch and bound method on the heterogenous environment

KATSUMI ONISHI, HIDEO NAKANO

#### 1. ま え が き

学術情報総合センター内のネットワークを利用して実験的なクラスタ計算機環境を構築し並列分枝限定法を適用した<sup>1)</sup> 報告を行った。その後、研究用情報処理教育システム<sup>2)</sup> を構成するシステムの一つとしてクラスタ型計算機システムが導入された。このようにクラスタ型並列計算機の利用はすすんでいるが、これらのシステムはいずれも単一機器で構成された環境であり複数の種類の機器で構成される環境についての利用も今後の検討課題である。本報告では二種類の計算機から構成される環境を用意しその上で並列分枝限定法を実行し単一機種環境との比較を行う。

分枝限定法は組合せ最適化問題の中でも特に難しい問題を解くために適用される解法として知られている<sup>4),5)</sup>。分枝限定法はどのような組合せ最適化問題にも適用しやすいという特徴を持っているが、最適解を得るために必要な計算時間は問題の規模が大きくなるにつれて指数関数的に大きくなる傾向がある。このため、分枝限定法を並列化し問題を高速に解くことが提案されている<sup>3),9)</sup>。

最近では、PC の上で動作する FreeBSD や Linux を利用したコンピューティング環境が比較的容易に構築できる。また、インターネット技術の発達によりネッ

トワークを通じて複数の計算機を接続し、一つの計算機資源として利用する試み(メタコンピューティング)がよく利用されている。このようなメタコンピューティング環境は従来の並列計算機と異なり、構成する計算機の能力が均一ではない。本報告では、2 種類の異なった計算機から構成される環境を取り上げ計算機の構成がかわることによる影響を並列分枝限定法を例にして検討する。

並列分枝限定法で解く具体的な組合せ最適化問題として巡回セールスマン問題を取り上げる。巡回セールスマン問題(TSP)は、組合せ最適化問題の中で、問題の記述の容易さ、応用の広さなどから特に広く研究されている問題であり<sup>6)</sup>、最適解を求めるアルゴリズムや、近似解を求めるアルゴリズムなどさまざまな解法が考えられている。

本報告では、複数機種環境での実行結果の報告を中心とし、分枝限定法およびその並列化、巡回セールスマン問題の詳細等には触れない。それらに関しては以前の報告<sup>1)</sup>を参照していただきたい。

#### 2. 並列分枝限定法

##### 2.1 分枝限定法

分枝限定法は、ほとんどすべての組合せ最適化問題に適用可能な計算手法である。分枝限定法では、直接解きたい問題を一部の変数の値を固定することでより解きやすい部分問題に分解する(分枝操作)。これらのすべてを解くことにより、間接的にもとの問題を

<sup>†</sup> 大阪市立大学学術情報総合センター  
Media Center, Osaka City University

解く。

可能な分解操作をすべて行うだけでは単純な列挙法と同じであり、大規模な問題に対して適用することはできない。これを避けるために不必要な部分問題の生成を可能な限り抑えることが必要である。分枝限定法では、探索中に利用できる情報を利用して不要な問題に対しては新たな分枝操作を適用しないようにする(限定操作)。分枝操作も限定操作もなされていない問題は活性部分問題とよばれる。処理すべき活性部分問題を探索木の中から選び出すことを探索とよぶ。探索の方法としては、分枝木上の位置に着目した探索(深さ優先探索、幅優先探索)、(対象とする問題が最小化問題の場合)部分問題の下界値に着目した探索、それらを組合せた探索などが考えられる。

## 2.2 仮想並列処理環境

本報告では、メタコンピューティング環境とも呼ばれるネットワーク上でワークステーションなどの計算機が複数接続されている分散処理環境を対象として並列分枝限定法のプログラムを作成する。

本報告では、並列処理システムとして星状型マルチプロセッサシステムを、PVM ライブラリを利用し、分散処理環境上に構築する。共有メモリは用意せず、各プロセッサは固有のメモリだけを利用する。また、プロセッサ間のデータの伝送はプロセッサ間でメッセージとしてネットワークを通じて伝送する。

## 2.3 並列分枝限定法のアルゴリズム

2.2節で述べたようなマルチプロセッサシステム上で並列分枝限定法を実現する方法として、本報告では次のような方法を採用している。

親プロセッサは与えられた問題のデータと解の管理、および子プロセッサへの問題の割当を行う。子プロセッサは親プロセッサから与えられた活性部分問題を初期問題とみなし、探索と分枝操作、限定操作を繰り返す。この方法では、親プロセッサに関しては記憶領域および通信時間が節約でき負荷は軽くなる。しかし、子プロセッサに割り当てる部分問題は、すべてが同じ時間計算量(手間)で終了するわけではなく、その違いをあらかじめ知ることができないため、各子プロセッサで探索終了時間に差が生じる。活性部分問題がなくなった空き状態の子プロセッサをそのままにしておくことは、計算機の利用効率の点から考えても明らかに良くないので、このような子プロセッサに対して、親プロセッサが再び部分問題を再割当する必要がある。

そこで、本報告では、子プロセッサから活性部分問題がなくなったという報告が来た場合、親プロセッサ

に用意するリストに活性部分問題があれば、ただちにその中から問題の一つを選び再割当を行う。割り当てべき活性部分問題がリストにない場合、その他の子プロセッサから、部分問題を一問ずつ集め親プロセッサのリストに登録し、その中から問題の一つを選び再割当を行う。

## 3. 計算機実験の方法と環境

### 3.1 分枝変数の選択

対称巡回セールスマン問題は、グラフの最小コスト巡回路を求める問題として定義され、定義からある枝を表す変数  $x_{ij}$  を“0”もしくは“1”にすることで部分問題に分割することができる。枝を選択する方法としては、まず距離の短い枝は巡回路に含まれやすいと思われるので枝をコスト順に選択する方法を利用する(Edge)。また、巡回セールスマン問題は、すべての都市を訪問する順序を並び替えその中からコスト最小の順列を求める問題とも考えられる。この場合、今までの訪問都市の中に含まれていない都市を次に選ぶことで部分問題を作る。この次に訪れる都市への枝を選択する方法を利用する(City)。

### 3.2 下界値と上界値

分枝限定法では、各部分問題の最適値に対する下界値と上界値の値が厳密に求められると、限定操作が効率良く行われ、分枝操作を必要とする活性部分問題の数が少なくなり、最適解を求めるために必要な時間を短くできる。このため、問題の性質を利用した精度のよい下界値・上界値を求めることが望まれる。

対称巡回セールスマン問題の下界値を求める方法として、最小1-treeの性質を利用した方法を用いる<sup>5),7)</sup>。1-treeとは、節点の数と同じ本数の枝から成り、無向グラフとして見た場合に連結しており、ある指定された一つの節点の次数が2であるようなグラフのことである。最小1-treeは、次数2の節点  $v_a$  を除く他の節点から成る最小木(minimum spanning tree)に、 $v_a$  に交わる枝の内で重みの軽いものから順に選んだ2本の枝を加えることにより求められる。

上界値としては、多くの組合せ最適化問題では最適解を得る必要がないならば、近似解法を用いて大規模な問題においても実用的な計算時間で近似解を求めることができる。この近似解法を用いて得られる初期問題の局所最適解を繰り返し複数個求め、それらの中で最も良い局所解を上界値として採用する方法を使用する。本報告では、近似解法として、近傍探索法の一つであるB.W.KernighanとS.Linの提案した解法<sup>8)</sup>を取り上げる。

表 1 Virtual machine の構成ホスト  
Table 1 Virtual machine

	A	B	C
機種	SONY QuarterL	PC	SUN SS20
OS	NEXT STEP 3.3J	FreeBSD 3.3	SUN OS4.1.4
メモリ	64MB	128MB	64MB
CPU	Pentium 100MHz	Celeron 433MHz	SuperSPARC 50MHz
台数	最大 51 台	最大 3 台	1 台

巡回セールスマン問題の例としては、TSPLIB\*と  
呼ばれる一般に公開されている問題例の中からいくつ  
かを選んで実行する。TSPLIBに含まれている巡回  
セールスマン問題の例は、この分野においてアルゴリ  
ズムの比較を行うために広く使われている。TSPLIB  
には、数都市程度の問題から数千都市程度の問題まで  
幅広い問題例が含まれているが、本報告では計算機を  
占有して利用できる時間の関係から、48都市 (GR48)、  
70都市 (ST70) の2題を取り上げ計算機実験を行った。

### 3.3 計算機環境

本報告において、PVMの仮想並列計算機を構成す  
るUNIXワークステーションとして、表1に示すよ  
うな計算機群を用いる。Aの計算機の中から1台を選  
び、親プロセッサ用のプログラムを実行する。親プロ  
セッサ用を除く他のA,Bの計算機上では子プロセッ  
サ用のプログラムを実行する。

上界値を得るための近似解法を実行するプログラム  
は別の計算機C用として用意し、他の親プロセッサ・  
子プロセッサとは独立に実行する。このプロセッサで  
求めた近似解は、限定操作の効率をよくするための暫  
定解として、他のプロセッサに放送される。

子プロセッサが次に分解すべき活性部分問題を選  
ぶ方法としては一般的により暫定解が早く見つかり、  
枝刈りの効率が上がるとされている最良下界値優先探  
索を各実験において共通の探索方法として採用する。

2.3節で述べたように、子プロセッサは、親プロセッ  
サからの要求に応じて他の子プロセッサに再割当され  
る部分問題を選ぶために子プロセッサ自身が持っている  
活性部分問題のリストの中を探索し、親プロセッサ  
に部分問題を送る。ここで親プロセッサに送る部分問  
題を探索する方法も負荷の分散を考える時には影響が  
ある。本報告では、この場合の探索方法については、  
幅優先探索を共通の探索方法とする。

## 4. 計算機実験の結果

単一の計算機で構成した環境での計算機実験の結果

をTSPLIBでの問題毎に図1(GR48)、図3(ST70)に  
示す。どちらの例題についても子プロセッサの台数が  
1台の場合から実験をはじめ、最大50台の子プロセッ  
サを利用して計算機実験を行った。グラフ中の左縦軸  
は、計算が終了するまでの経過時間を秒単位で表して  
いる。また、グラフ中の右縦軸は、子プロセッサが2  
台の場合を基準にして利用した台数に見合う加速度が  
得られるかを表示している。 $n$ 台の子プロセッサを利用  
すれば理想的には $n$ 倍の加速度が期待されるが、グ  
ラフからもわかるとおり通信のオーバーヘッドや負荷  
の不均一な分散などの原因により $n$ を下回る加速にな  
っている。グラフ中には、経過時間(時間)・加速度  
(加速)でそれぞれ2つの結果がある。それぞれの設定  
の違いは分枝変数の選択方法の違いによって異なる。  
分枝変数の選択方法は、3.1節で述べたように枝をコ  
スト順に選択するか(Edge)、訪問都市(City)をもと  
に選択するかである。

次に、2種類の計算機による結果を図2、図4に示  
す。これらの図では、Bの計算機を1,2,3台と利用し、  
各場合についてAの計算機を1台から20台利用した  
結果を経過時間と加速度に関して問題例毎に2つのグ  
ラフで示している。まず、性能の異なる計算機として  
用意した1台の計算機Bは6.3台の計算機Aに相当  
することをGR48,EIL51,ST70,RAT99を1台の計算  
機Aで解いた結果との比較から確認した(表2)。表2  
では、それぞれの問題を2つの分枝方法で解く場合の  
実行時間(秒)を計算機Aが1台、計算機Bが1台  
の場合についての結果をまとめ各問題例毎にその比率  
も比較している。図の中ではこの結果の平均である1  
台の計算機Bが6.3台の計算機Aに相当するとして  
グラフ中の横軸を設定している。加速度の計算に関し  
ても同様の換算を行った。つまり、計算機B1台と計  
算機Aの組合せは図2(GR48b)、図4(ST70b)にお  
いて計算機台数が7.3台から26.3台の部分に相当する。

単一の計算機での結果と複数の計算機での結果を比  
較する。まず、問題を解くための経過時間を見る(図  
1, 2(GR48)、図3, 4(ST70))。同じ性能構成の範囲  
で比べるとどちらの問題でも同じ設定であれば同じ程

\* ftp://elib.zib-berlin.de/pub/mp-testdata/tsp/

表 2 構成ホスト A,B の性能  
Table 2 Performance of computer A, B

問題		B,1 台	A,1 台	B/A
GR48	(都市)	410.5	2575.01	6.27
	(距離)	194.79	1198.52	6.15
EIL51	(都市)	171.74	1051.58	6.12
	(距離)	67.56	430.20	6.37
ST70	(都市)	3663.66	25251.90	6.89
	(距離)	5007.05	30861.50	6.16
RAT99	(都市)	542.7	3408.7	6.28
	(距離)	631.31	3838.6	6.08
平均				6.29

(単位:秒)

度の経過時間で問題を解くことが確認できる。2種類の設定間の優劣に関して比較してもどちらの構成でも同じ傾向になっている。次に加速度について比較する。ここでの加速度は先に述べたように計算機 B の能力を 6.3 台の計算機 A に相当するとして計算している。利用台数が少ない範囲では同じような結果にまとめられる。しかし、性能差が大きい計算機が含まれることで全体としての結果が悪くなるような事はないことがわかる。また、台数が多くなるに連れて単一環境の場合と比較して加速度の劣化が低く押さえられている。これは、性能の良い計算機が多くの部分問題を担当することで計算機間の通信が少なくなることから無駄な時間が押さえられるためであると考えられる。

このように並列分枝限定法を異機種で構成された計算機環境で実行する場合、構成する計算機の性能の差による影響は少ないことがわかる。これは並列分枝限定法では、各部分木を割り当てられた子プロセッサが他の子プロセッサとは独立に探索操作を行うためであると考えられる。特に本報告では、部分問題がなくなった子プロセッサに再割当する問題として各子プロセッサから幅優先探索により分枝木上で根に近いできるだけ大きな部分木を選択するため、再割当の回数を低く押さえることができたためであるとも考えられる。再割当が頻繁に起こるようであれば各子プロセッサが独立に探索する割合が少なくなるため、加速度等が悪くなるが考えられる。

### 5. む す び

本報告では、メタコンピューティング環境とも呼ばれる複数の計算機がネットワークを介して接続された分散処理環境を想定し、具体的な問題として巡回セールスマン問題を並列分枝限定法を用いて解くことを取り上げた。特に計算機環境として単一の計算機で構成された環境だけではなく、性能差のある 2 機種から構成された環境で実行しその結果を比較・検討した。

二つの問題例を取り上げ、分枝限定法を実行する場

合に重要な分枝変数の選択方法を変更することも組み合わせさせて比較を行った。経過時間・加速度いずれの比較についても性能差がある計算機を組み合わせた事により結果が悪くなる事はなかった。並列分枝限定法では各子プロセッサが独立に探索を進められるから再割当をうまくおこなう事により同じような性能構成でも加速度が向上する場合があった。

今後の課題としては、本報告では、LAN 環境を前提とし子プロセッサ間を接続するネットワークに特に遅延等が生じる事は考慮に入れていない。そのため、広域のネットワークで遅延が生じるような場合の評価が考えられる。また、子問題の評価時間が異なるような場合を含め TSP 以外の組合せ最適化問題への適用等があげられる。

謝辞 このクラスタ型並列計算機システム構築に際して、ご協力頂いたシステム管理系の職員の方々、教育用計算機システムの運用担当の先生方に謝意を表します。

### 参 考 文 献

- 1) 大西克実, 中野秀男 “PC クラスタ環境の構築と並列分枝限定法への適用”, 大阪市立大学学術情報総合センター紀要, vol.2, pp.25-30, 2001.
- 2) 中野秀男, 大西克実 “研究用情報処理システムの新機能” 大阪市立大学学術情報総合センター紀要, vol.3, pp.15-25, 2002.
- 3) 今井正治, 吉田雄二, 福村晃夫: “分枝限定アルゴリズムの並列化とその評価”, 信学論, J62-D(6), pp.403-410, 1979.
- 4) 茨木俊英: “組合せ最適化 分枝限定法を中心として”, 講座・数理計画法第 8 巻 産業図書, 1983.
- 5) 今野浩, 鈴木久敏: “整数計画法と組合せ最適化”, OR ライブラリー第 7 巻 日科技連出版社, 1982.
- 6) E.L.Lawler and J.K.Lenstra and A.H.G.Rinnooy Kan etc.: “The travelling salesman problem”, John Wiley and Sons, 1985.
- 7) M.Held and R.M.Karp: “The Traveling Salesman Problem and Minimum Spanning Trees: Part II”, Math.Progm, 1, pp. 6-25, 1971.
- 8) S.Lin and B.W. Kernighan: “An efficient heuristic algorithm for the traveling salesman problem”, Operations Research, 21, pp. 498-516, 1973.
- 9) Catherine Roucariro: “Parallel branch and bound algorithms - an overview”, In M.Cosnard et al., Parallel and Distributed Alogorithm, pp. 153-16, North-Holland, 1989.

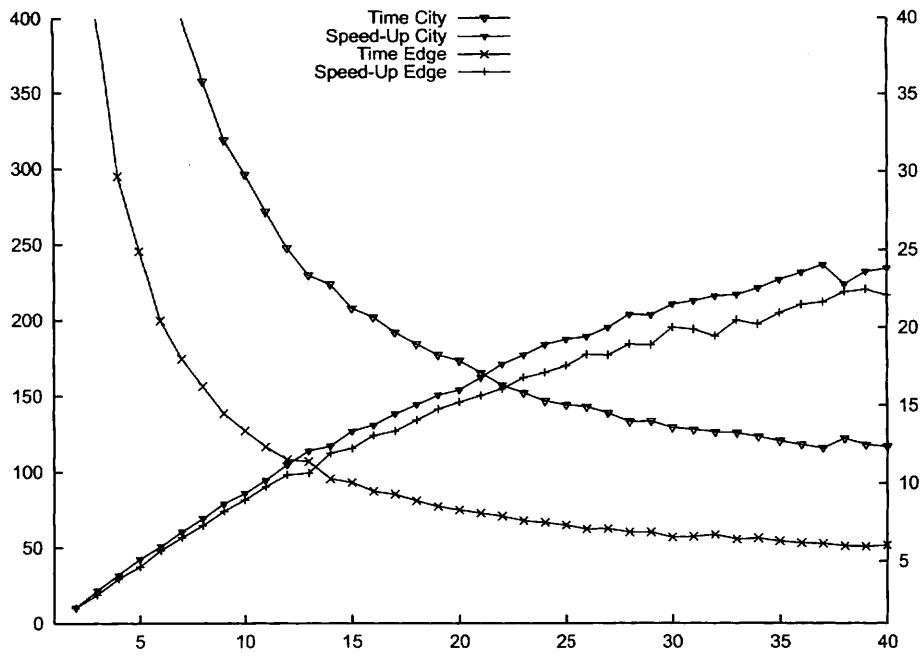


図 1 計算時間と加速度 (GR48)

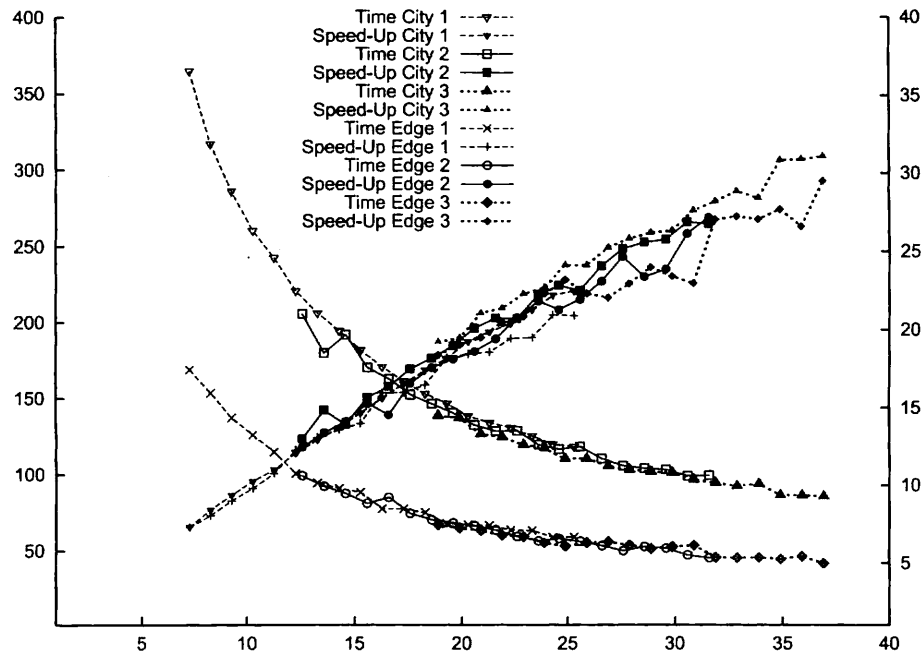


図 2 計算時間と加速度 (GR48b)

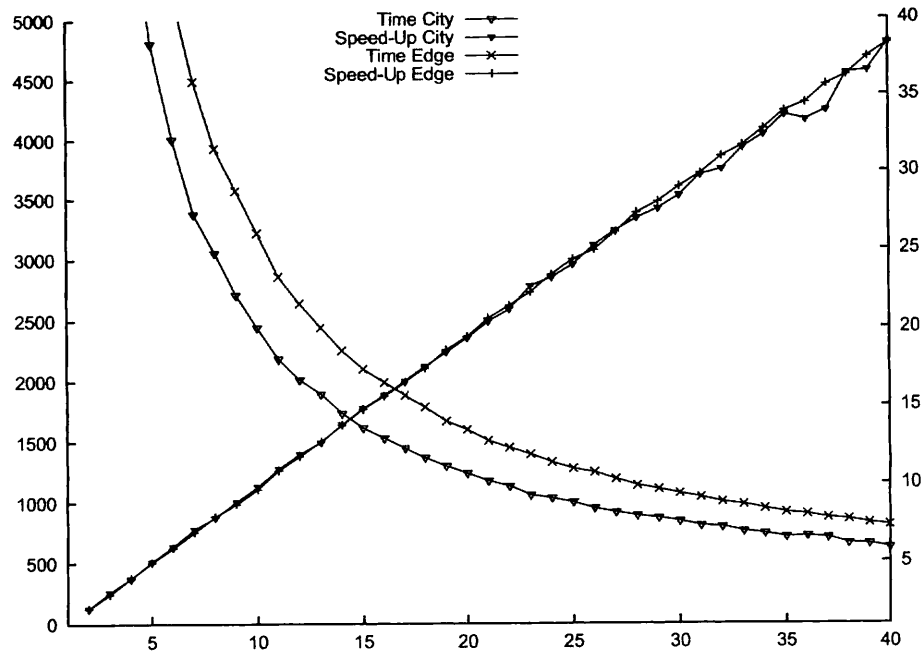


図 3 計算時間と加速度 (ST70)

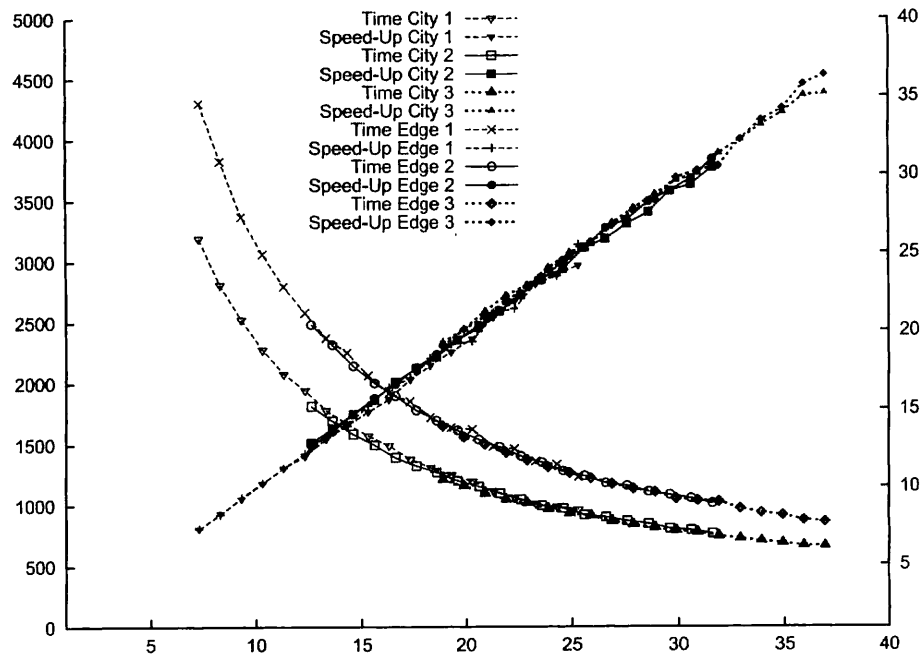


図 4 計算時間と加速度 (ST70b)