

平成 23 年度修士論文

災害初期の啓開における  
地域間連携の有効性の検討

渡邊 裕介

平成 24 年 1 月 30 日

東京都市大学大学院 工学研究科 都市基盤工学専攻  
計画マネジメント・皆川研究室

# 目次

## 第1章 緒言

背景

## 第2章 既往の研究と本研究の目的

2.1 既往の研究

2.2 本研究の目的

## 第3章 シミュレーション概要

3.1 想定するシナリオ

3.2 シミュレーションに用いる要素

3.3 エージェントの行動ルール

## 第4章 シミュレーション結果

4.1 静岡県全域での連携を考慮したモデル

4.2 静岡県内4地域内での連携を考慮したモデル

4.3 緊急輸送路の被害を想定したモデル

4.4 市町村別のがれき処理量、稼働重機数の時間別推移

4.5 各モデルによる結果の比較

## 第5章 結論

付録

参考文献

## 第 1 章

### 緒言

## 背景

地震国である我が国では、従来の地震・台風・洪水等により多くの災害を経験してきたことから、地方自治体ではそれらの災害に適切に対応するために、地元の建設業者と適宜災害協定を結んでおり、災害発生時の迅速な対応、復旧を行う上で重要な役割を果たしてきた<sup>1)</sup>。特に、平成23年3月11日に発生した東日本大震災は、1000年に一度という未曾有の地震規模とそれによる津波が多大な被害をもたらした。この際、東北道・国道4号線から、津波で多大な被害を受けた三陸沿岸部へ進出して復旧活動を行えるよう、「くしの歯型」救援ルートが東北地方整備局により設定され、地元建設業者の迅速な協力により、発災後わずか4日間で15ルートのすべてが通行可能となった<sup>2)</sup>。この例からも改めて示されたように、緊急に救援ルートを確保する啓開活動、及びそれに付随する被災者捜索活動において、県外からの大手建設業者の支援は困難であり、地元建設業者による対応が重要となる。

一方、談合からの決別と総合評価落札方式などの新しい制度の導入により競争は激化しており、図-1に示す過去14年間の公共事業費の推移に見られる急激な公共事業費抑制抑向などが影響して、地元建設業者は急激に弱体化している。図-2に我が国の建設事業所が保有する建設重機数の推移を示す。この結果から、地方の建設業者の建設重機の保有台数もまた減少していると見れる。

このような状況のなかで、地域防災力の向上が求められる。今後予想される大規模な災害として、東海地震、東南海地震、南海地震などがある。東海地震を例にとると、M8クラスの地震が発生し、死者は約7900～9200人、全壊家屋は約23～26万棟との想定結果が出されており、図-3に示すような大きな揺れにより大規模な被害が予想される。災害初期の数日間は生存率に大きく関わり、この数日間の救出活動は重要である。このような震災が発生すれば建設重機の不足は啓開及び被災者救出活動に甚大な影響を及ぼすと考えられる。

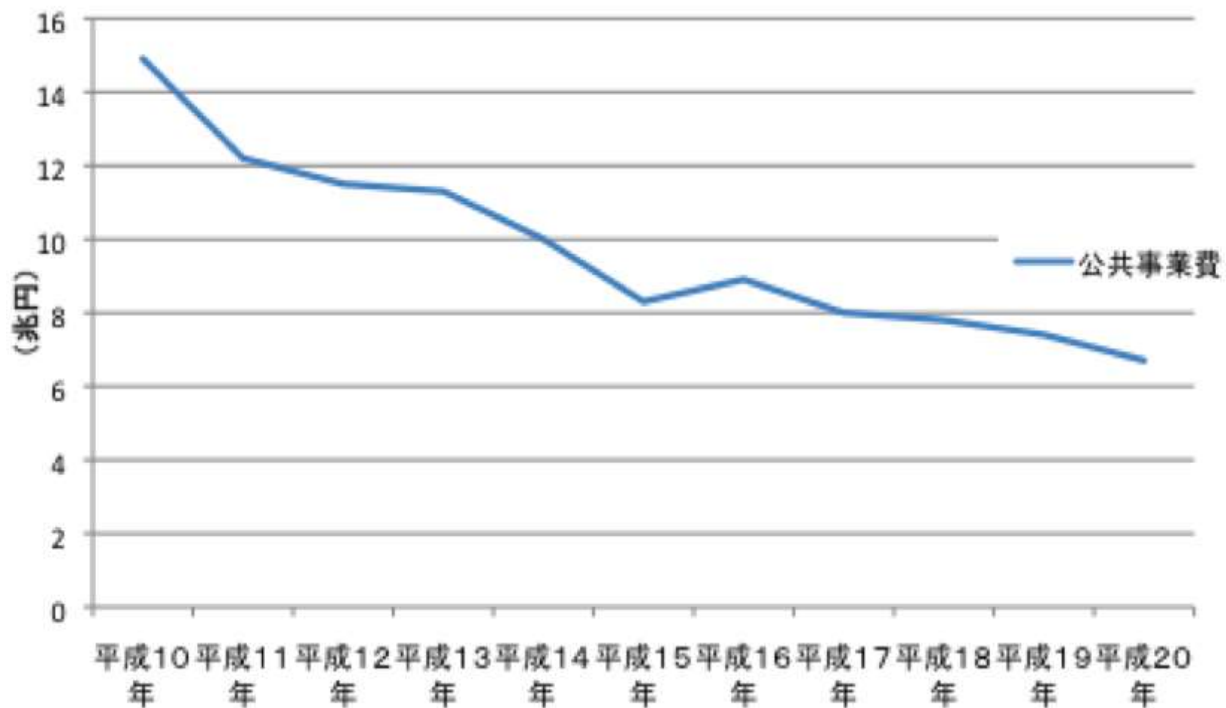


図-1 公共事業関連予算の推移<sup>3)</sup>

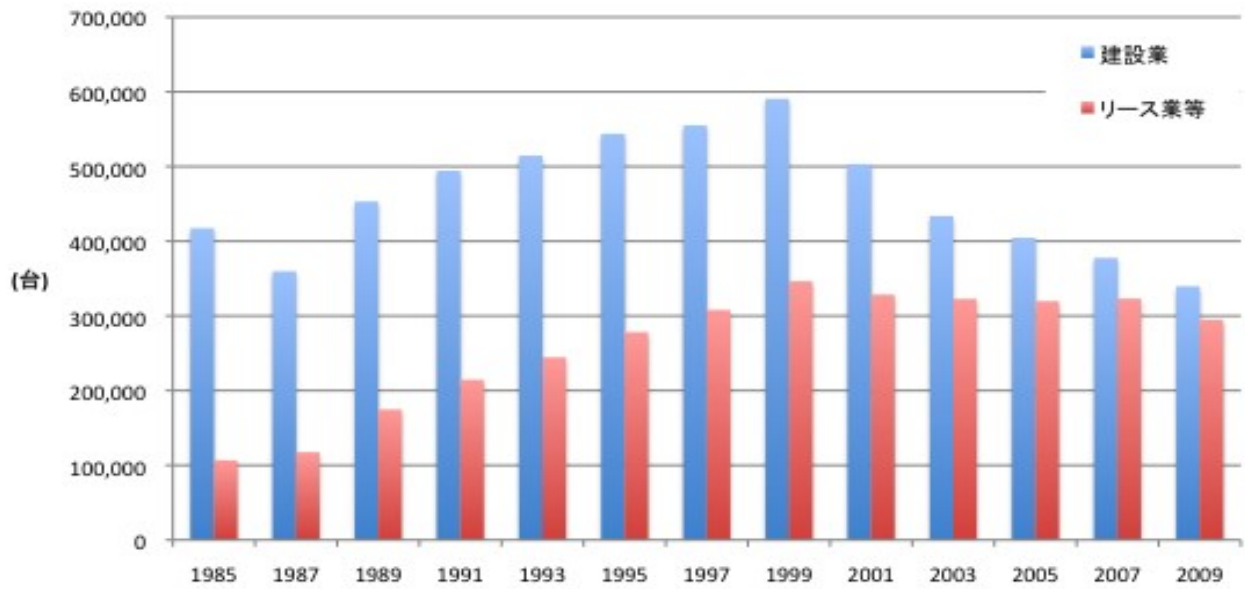


図-2 建設重機保有台数の推移

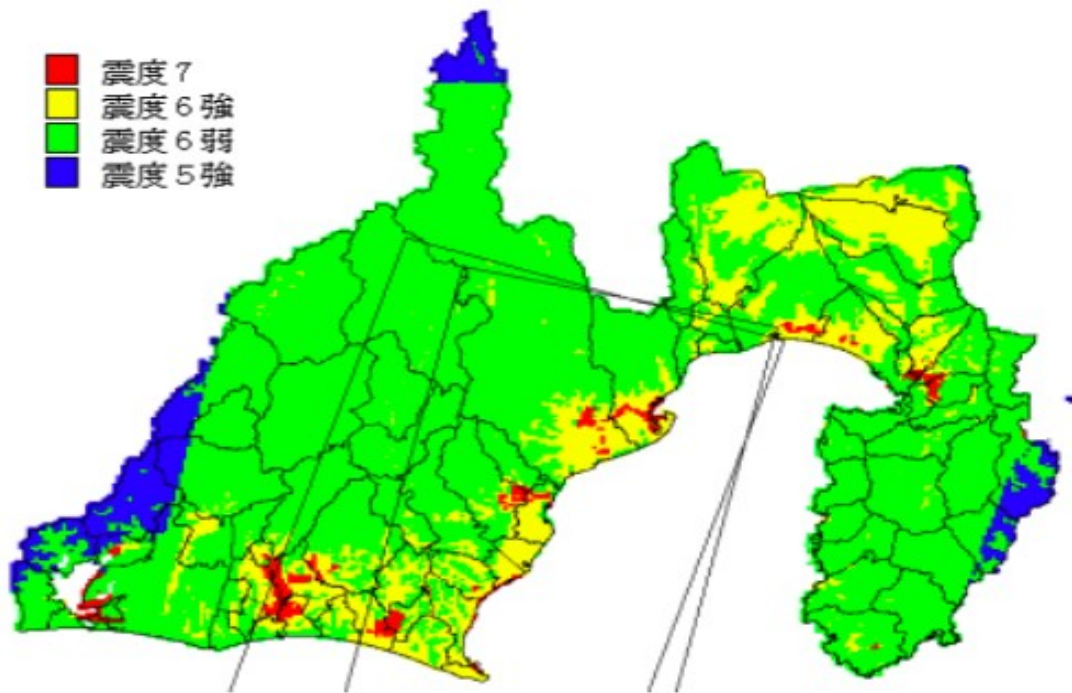


図-3 東海地震震度分布<sup>5)</sup>

## 第 2 章

### 既往の研究と本研究の目的

## 2.1 既往の研究

### a) マルチエージェントシステムを用いた研究

ある事象を構成する要素同士の相互作用やそれに伴う現象の再現に適したシステムとして、マルチエージェントシステムがある。マルチエージェントシステムは一般的にエージェントと呼ばれる多数の自立した個々の主体が存在し、それらの相互依存作用により成り立つシステムである。マルチエージェントシステムは、事象全体をモデル化するシミュレーション方式と異なり、事象を構成する基本的な要素をモデル化することにより、それによって引き起こされる事象全体を再現するものである。近年、表-1に示すようにマルチエージェントを用いた災害に関する研究が数多く行われている。このように、被災者の避難行動を扱った研究が多いが、その他にも被災時の車両の運行を扱った研究もある。

表-1 マルチエージェントを用いた防災研究

著者	年	タイトル	成果
宮嶋ら <sup>6)</sup>	2007	多様な群集の雑然とした状況を想定した地震時避難行動シミュレーション	地下鉄駅を対象に避難行動シミュレーションを行い、避難行動の差を定量的に分析した。
笹岡ら <sup>7)</sup>	2011	災害現場における最短経路探索システムの構築	迅速な災害救助を実現する為に、ロボットとマーカーを配置したシステムを提案した。
前地ら <sup>8)</sup>	2010	DIGへの適用を目的としたマルチエージェント避難行動シミュレータの試験的開発	MASを使った延焼モデル、道路閉塞モデル、避難行動モデルを開発し、DIGの現場での活用実験でその有用性を検証した。
野澤ら <sup>9)</sup>	2005	マルチエージェントシステムを用いた歴史的市街地における津波避難シミュレーションモデルの構築	津波来襲時の避難行動をモデル化し道路閉塞の影響や対策の違いを避難者数の変化で示すことができた。
村木ら <sup>10)</sup>	2004	マルチエージェントモデルを用いた広域災害避難シミュレーションにおける情報伝達の有効性	広域災害避難シミュレーションを行うためのマルチエージェントモデルを構築した。

## b)建設重機と地域防災力に関する研究

馬場ら<sup>1)</sup>は、高知県を対象に、地元建設業と地域防災に関する研究を行った。川上による地域防災力の試算式<sup>2)</sup>を用い、近い将来発生するとされている南海地震時に想定されている家屋倒壊の撤去と生き埋め者の救助に要する日数について試算を行い、表-2の結果を得ている。この試算の結果、高知県内での建設重機の偏在により高知市における所要日数が県全体の所要日数の2倍と大きくなることが予測され、県内の各地域内での建設業者のみの対応では特定の地域のがれき処理と被災者捜索が遅れることが分かる。このように、建設業者が保有する建設重機の偏在が地域防災力へ影響を与えることが予想されるが、建設重機と防災に関する既存の研究は少ない。

表-2 建設重機台数による地域防災力の試算結果<sup>7)</sup>

高知県における試算結果			高知市における試算結果		
	重機数	所要日数		重機数	所要日数
H16	1277	4.9	H16	122	12.3
H17	1254	5.0	H17	109	13.8
H18	1136	5.5	H18	88	17.1
H19	1060	5.9	H19	94	16.0
H20	762	8.2	H20	93	16.2

$$\text{救出日数} = \frac{\text{倒壊家屋数 (全壊)}}{\text{重機数} \times \text{重機 1 台 1 日 当 たり の 仕 事 量}}$$

## 2.2 本研究の目的

以上の事から、本研究では、地域防災力の要因のひとつである建設業者が保有する建設重機数に着目する。東海地震の発生を想定して対象地域を静岡県とし、県内の市町村別の建設重機数を調査し、静岡県が作成した第3次被害想定<sup>5)</sup>をもとに地域間での災害初期における重機の共有体制の有無が啓開においてどのような影響を及ぼすかを、建設重機をエージェントとしたマルチエージェントシミュレーターを用いてシミュレートする。これにより、震災発生時の地域連携における効率的な重機の共有体制が災害初期活動に及ぼす効果を把握する。尚、啓開に関し、建物の倒壊による生き埋め者の救助も考慮するために、道路上のがれき処理に留まらず、市町内の全がれき処理を想定した。



## 第 3 章

### シミュレーション概要

### 3.1 想定するシナリオ

東海地震により、第3次被害想定<sup>5)</sup>で示されている被害が発生してがれき及び生き埋め者の救助が必要となった状況を想定する。第3次被害想定<sup>5)</sup>より静岡県内の各市町村で発生するがれき量と、静岡県との災害協定により地元建設業者が災害時の提供に同意している建設重機数を設定する。重機は緊急輸送路を用いて市町間を移動するが、一般車両による渋滞は考慮しない。重機の作業能力は川上の試算式を用いて作業効率を設定して各市町でのがれき処理作業を行い、県内全てのがれきの処理が完了するのに要する時間を測定する。想定するシナリオを以下に示す。

- 1) 静岡県全域ですべての建設業者が連携してがれき処理を行う場合（全域連携）
- 2) 静岡県を西部，中部，東部，伊豆の4地域に分割し，その域内のみで建設業者が連携してがれき処理を行う場合（地域連携）
- 3) 富士川付近の緊急輸送路が被災して早期の復旧が見込めなく，県内をその東西の2地域に分けて，それぞれの域内で建設業者が連携してがれき処理を行う場合（分断地域内連携）
- 4) 各市町内で建設業者ががれき処理を行う場合(単独)

### a) 市町別建設重機数

静岡県内の建設重機数に関して調査を行った。建設重機に関するの先行研究はほぼ存在せず，静岡県交通基盤部への問い合わせに対して同部ではその重要性に鑑みて，市町別の災害協定に基づく災害派遣用の建設重機数を調査した。表-3に同部より提供された重機・オペレーター保有数の集計結果<sup>11)</sup>を示す。このデータは暫定的なものであるものの，現在，静岡県内の重機数が記載されたデータは他に確認されておらず，現段階では最も信頼性の高いデータであると考えられる。ここで，シミュレーションに用いる重機は汎用性があり，重機のシェアの大半を占めているバックホウとしている。

表-3 静岡県内市町別バックホウ保有台数<sup>11)</sup>

市町名	バックホウ類 (台)	市町名	バックホウ類 (台)	市町名	バックホウ類 (台)	市町名	バックホウ類 (台)
下田市	33	伊豆の国市	100	富士宮市	97	磐田市	41
東伊豆町	14	函南町	19	富士市	110	掛川市	85
河津町	31	沼津市	70	静岡市	352	袋井市	39
南伊豆町	22	三島市	14	島田市	202	御前崎市	82
松崎町	25	御殿場市	34	焼津市	127	菊川市	29
西伊豆町	38	裾野市	27	藤枝市	180	森町	13
熱海市	35	清水町	17	牧之原市	115	浜松市	419
伊東市	50	長泉町	43	吉田町	33	湖西市	33
伊豆市	118	小山町	18	川根本町	68	県合計	2733

## 重機数

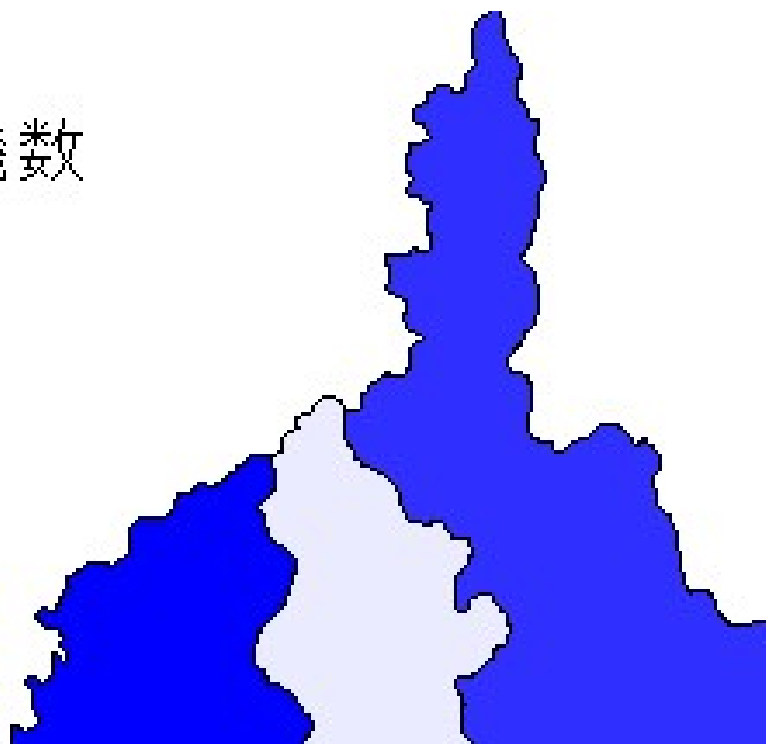


図-4 静岡県内市町村別バックホウ保有台数<sup>11)</sup>

## b) 市町別がれき推定発生量

静岡県が発表した第3次被害想定<sup>5)</sup>より、市町別の倒壊家屋数が決定される。第3次被害想定には市町村別のがれき推定発生量は記載されていない。そこで、まず、中部経済連合会が2008年に発表した「大震災に備えた震災がれき処理について」<sup>12)</sup>より静岡県内全域でのがれき推定発生量を決定した。そして、市町別の倒壊家屋数をがれき推定発生量の分布と同一と仮定し、市町別の推定がれき発生量を表-4に示すように割り振った。

表-4 市町別がれき推定発生量<sup>12)</sup>

市町名	がれき発生量(m <sup>3</sup> )	市町名	がれき発生量(m <sup>3</sup> )	市町名	がれき発生量(m <sup>3</sup> )	市町名	がれき発生量(m <sup>3</sup> )
湖西市	29.78	吉田町	31.83	小山町	11.07	伊豆市	17.04
浜松市	506.47	島田市	87.64	裾野市	12.45	伊東市	33.51
磐田市	112.26	藤枝市	143.80	長泉町	14.63	西伊豆町	10.16
袋井市	130.87	焼津市	162.71	沼津市	278.81	松崎町	12.15
森町	16.37	静岡市	1166.79	清水町	32.90	河津町	4.07
掛川市	75.82	川根本町	7.69	三島市	160.65	東伊豆町	4.76
菊川市	61.55	富士宮市	101.53	函南町	38.20	下田市	35.96
御前崎市	31.94	富士市	168.38	熱海市	15.72	南伊豆町	9.07
牧之原市	61.26	御殿場市	28.31	伊豆の国市	64.94	県合計	3681.08

## 推定がれき発生量

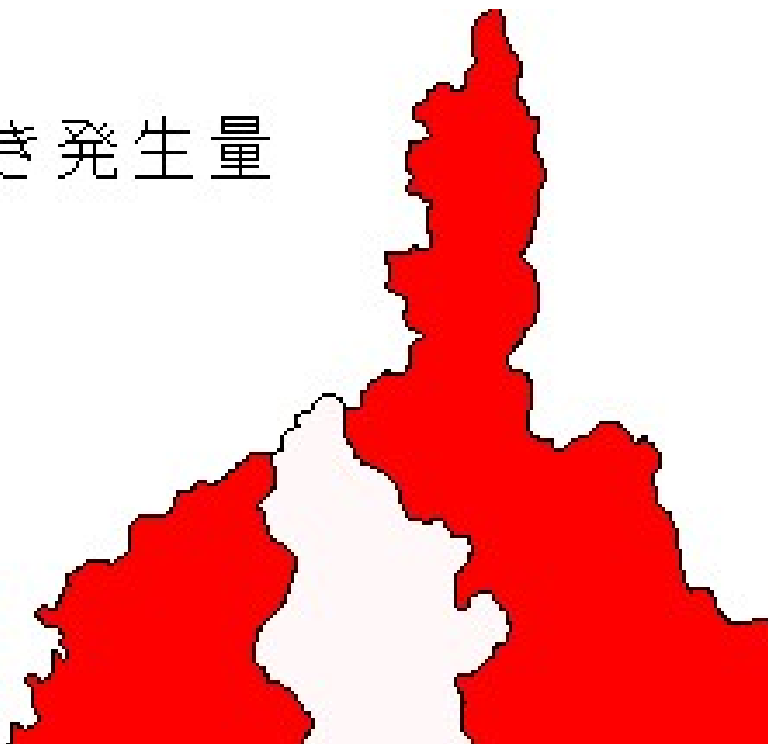


図-5 市町別がれき推定発生量<sup>12)</sup>

c) 建設重機のがれき処理能力

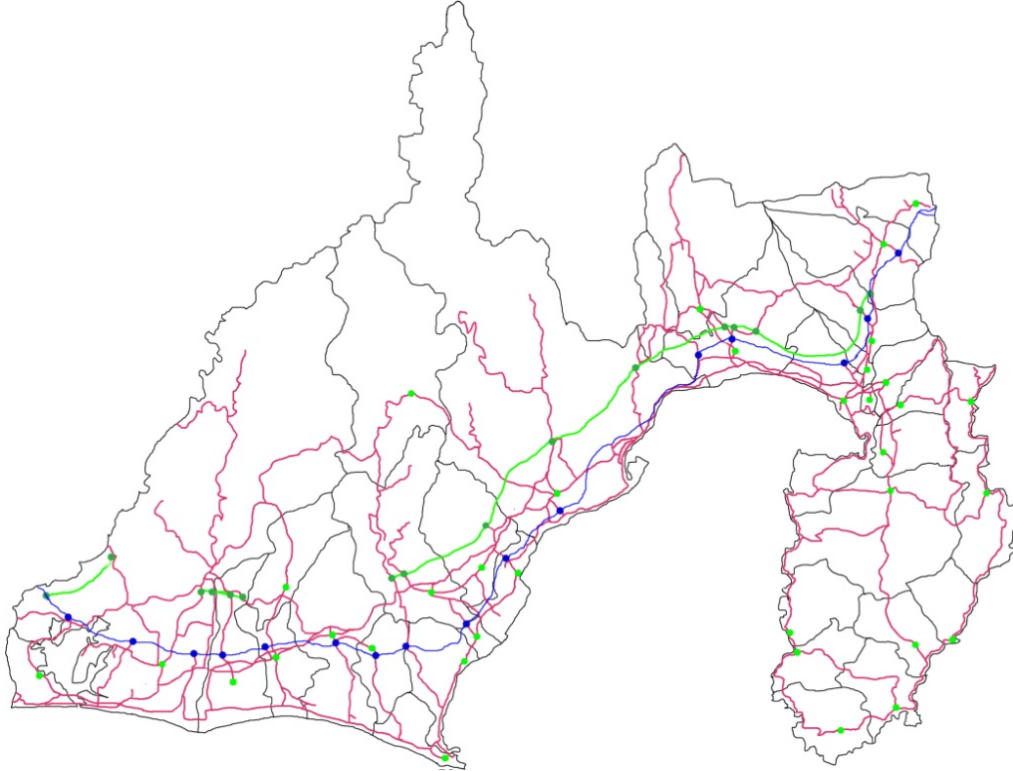
建設重機の車種は先でバックホウに限定した。バックホウのがれき処理能力は国土交通省が定めている土木工事標準積算書<sup>13)</sup>をもとに決定される。表-5に示す処理能力に従い、震災発生時の活動を想定するために劣悪な作業環境が予想され、作業の種類はルーズな状態の積み込み、バックホウの規格に関しては、市町保有の個々のバックホウ規格に関する詳細な調査する事が困難であるために全ての規格の平均の値を用いる事とする。重機の作業時間は労働基準法に基づき1日8時間とされているが、震災時の想定を行うために1時間あたりの作業量を算出し、24時間態勢での作業とする。

表-5 バックホウの処理能力<sup>13)</sup>

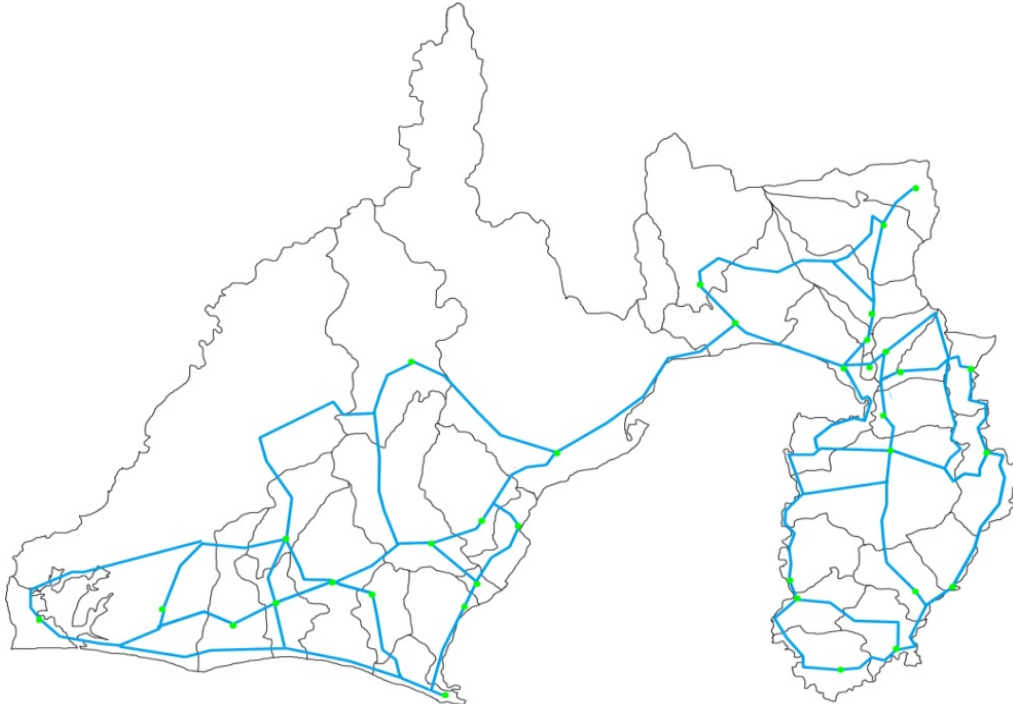
作業の種類	名称	規格	土質名	単位	障害なし	障害あり
地山の掘削積込	バックホウ運転	排出ガス対策型（第2次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	300	190
			岩塊・玉石	m <sup>3</sup>	230	140
		排出ガス対策型（第1次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	500	320
			岩塊・玉石	m <sup>3</sup>	410	260
ルーズな状態の積込	バックホウ運転	排出ガス対策型（第2次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	310	
			岩塊・玉石	m <sup>3</sup>	260	
		排出ガス対策型（第1次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	520	
			岩塊・玉石	m <sup>3</sup>	440	
		排出ガス対策型（第1次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	160	
			岩塊・玉石	m <sup>3</sup>	130	
床掘り（作業土工）	バックホウ運転	排出ガス対策型（第2次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	220	180
			岩塊・玉石	m <sup>3</sup>	160	130
		排出ガス対策型（第1次基準値）クローラ	砂・砂質土・レキ質土・粘性土	m <sup>3</sup>	150	100
			岩塊・玉石	m <sup>3</sup>	110	70

#### d) 緊急輸送路

震災時に通行可能な第3次被害想定<sup>5)</sup>で指定された緊急輸送路のデータを用いて、シミュレーション用の道路網を図-6(a)のように作成した。実際にはシミュレーションの負荷軽減のため、このネットワークを図-6(b)のように簡略化し使用する。



(a) 第3次被害想定<sup>8)</sup>で指定された緊急輸送路



(b) 簡略化した道路ネットワーク

図-6 緊急輸送路

e) 建設重機の初期配置 およびがれき発生地点

本研究では広域での市町間の連携を重視している。そのために市町内でのミクロな動きは無視し、市町内での建設重機の初期配置地点およびがれき発生地点は各市町の役所とする

f) 建設重機移動速度

建設重機の移動速度は国土交通省発表の道路交通センサス<sup>14)</sup>より混雑時旅行速度35.1km/hとする

g) 優先市町の設定

推定発生がれき量に対して、保有重機数が極端に少なく、がれきの処理に多大な時間を要する地域を優先地域とした。がれき発生量に対する重機台数の比率が0.5以下と小さく、優先地域として選定した市町を表-6, 図-7に示す。

表-6 優先市町

市町名	がれき発生量(m <sup>3</sup> )	バックホウ数(台)	重機比率
磐田市	112.26	41	0.37
袋井市	130.87	39	0.30
菊川市	61.55	29	0.47
静岡市	1166.79	352	0.30
沼津市	278.81	70	0.25
三島市	160.65	14	0.09
函南町	38.20	19	0.50



図-7 優先市町の重機比率

### 3.3 エージェントの行動ルール

通常の移動を行う通常エージェント及び、優先作業地域へ向かう優先エージェントの2種類の重機を表すエージェントを用意する。通常エージェントは図-8、優先エージェントは図-9に示すフローチャートに従い行動する。

優先エージェントの割合はパラメーターとして、ある一定の割合で各市町に発生させるのとする。通常の移動を行うエージェントは、まず、初期配置された市町内でのがれき処理作業を行う。作業完了後、ダイクストラ法を用いて最短距離で移動可能な他の市町を選定し移動する、全ての市町で作業が完了するまでこの移動と作業を繰り返す。

一方、優先エージェントに関しては、移動先を優先市町とし通常エージェントと同様の行動を繰り返す。優先地域での作業がすべて完了した後、通常エージェントに置き換え、他の通常エージェントと共に作業を行う。すべての地域で作業が完了した時点でシミュレーションは終了する。

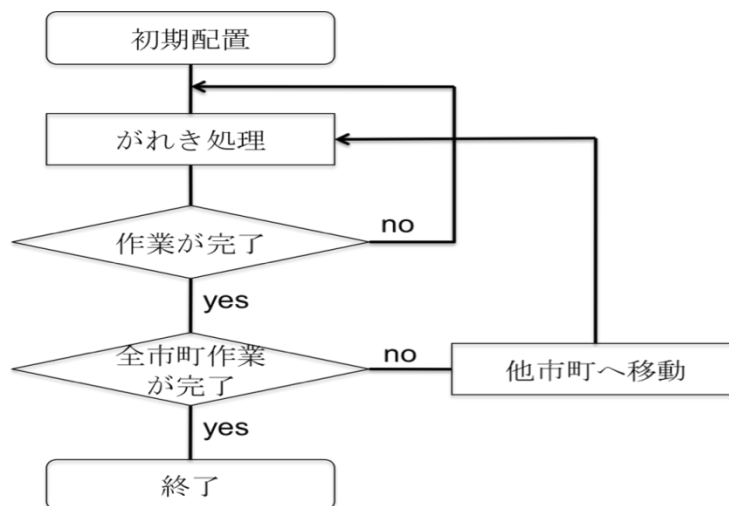


図-8 通常エージェントの行動

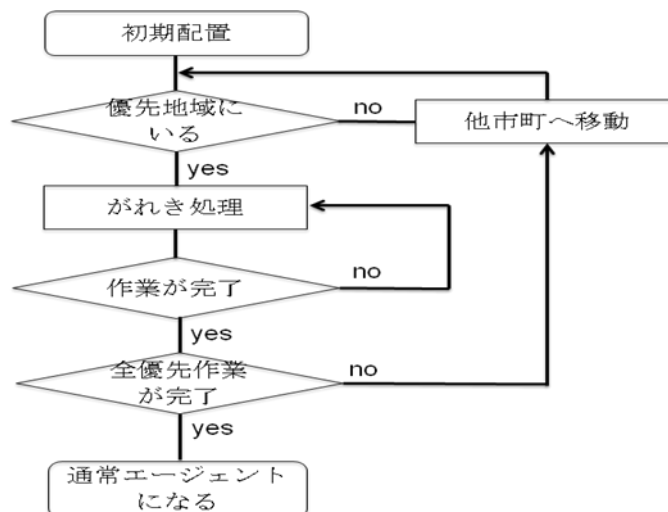


図-9 優先エージェントの行動



## 第 4 章

### シミュレーション結果

#### 4.1 静岡県全域での連携を考慮したモデル

静岡県全域ですべての建設業者が連携してがれき処理を行う場合を想定したシミュレーションを行った。

まず、優先エージェントがない場合の、市町別のがれき処理日数を図-10及び表-7に示す。表-8には、4.2で述べる地域連携処理日数及び、4.3で述べる分断地域内連携処理日数も合わせて示している。図中には、シミュレーションの結果及び市町内でそれぞれ単独にがれきを処理した場合の結果を合わせて示した。全ての市町において全域での連携による処理日数（全域連携処理日数d）と呼ぶ。）が、単独での処理日数（以後、単独処理日数a）と呼ぶ。）を下回る結果となり、広域による連携の効果を把握する事ができた。

シミュレーションにおいて最も遅く処理が完了した袋井市の日数が14.89日であるのに対して、県内の全がれきを利用可能な全重機で処理した場合の所要日数は14.80日となり、わずかにシミュレーション値が大きくなる結果となった。この増加の要因は重機の移動時間である。今回のシミュレーションにおいては重機の移動については渋滞を考慮せず、また多台数での重機の移動についても単独での移動と同じ時間としたことにより、移動に要する時間はがれき処理に要した時間に比べて小さい。ここでは、全域での連携という理想的な状況を想定しているため、効率的な作業が行われ、このような結果になったと考えられる。

次に、優先エージェントの割合（以後、優先割合と呼ぶ。）をパラメーターとし、0～50%の間で、10%ごとに変化させ、同様のシミュレーションを行った。重機の優先派遣に関しては、表-6で示した優先市町について、優先割合と市町ごとの処理日数の関係を図-11、表-7に示す。図中には、県内全域の処理日数も示した。同図から分かるように、三島市では明らかな傾向は示されていないが、他のすべての優先市町で処理日数が減少した。県内全域の処理日数は、優先市町に重機が回ったことによりわずかに増加しているものの、多くの優先市町で処理日数が減少している事から、優先市町への効率的な重機の派遣が行えたと言える。三島市において一時的に処理日数が増加している原因としては、周辺に他の優先市町が密集したために、優先エージェントがそちらへ移動した事が挙げられる。また、重機の移動履歴を図-12に示した。初期配置地点からの移動を1次とし、1次移動先からの移動を2次移動とした。3～7次の移動に関しても同様とした。低次での移動は近距離が多い。高次の移動になるに従い、静岡県の東側から西側に向けて多くの重機の移動が見られた。

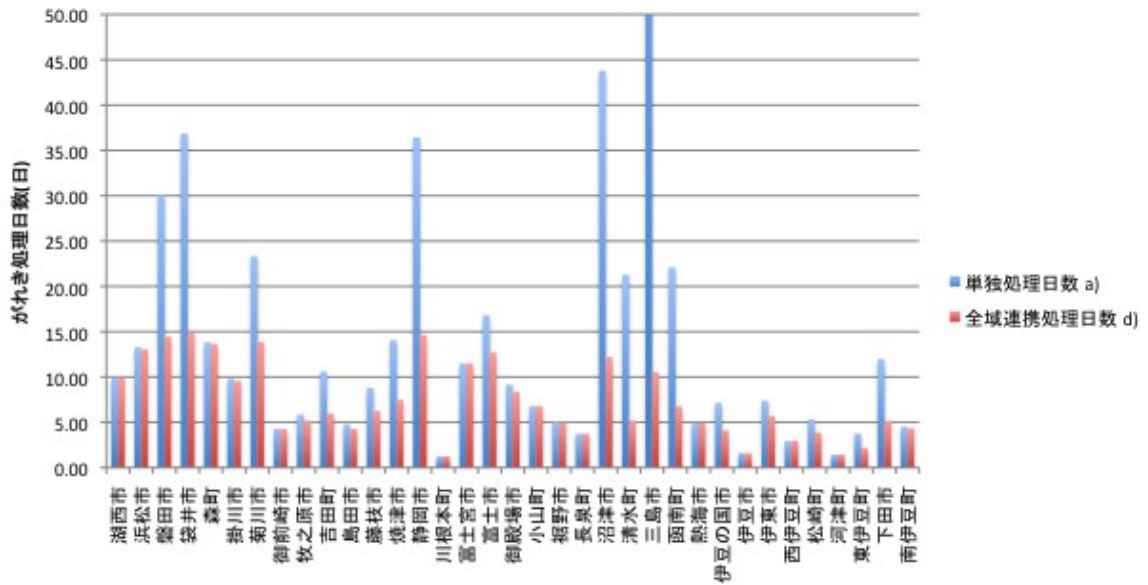


図-10 全域連携処理日数と単独処理日数の比較

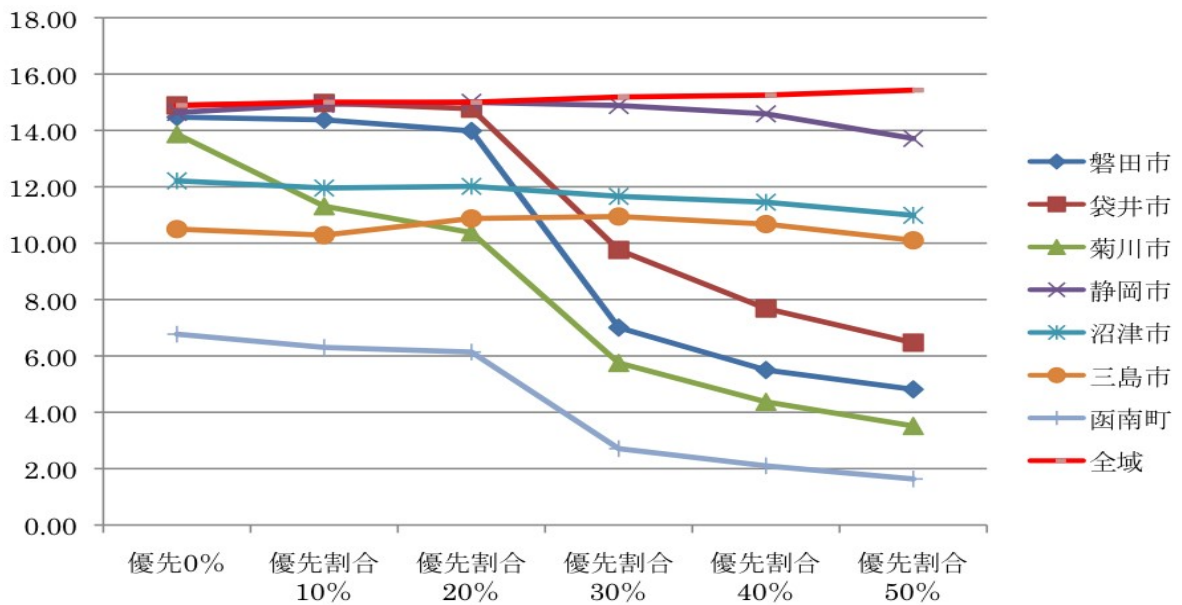


図-11 優先割合による市町ごとの処理日数の推移

表-7 全域連携, 優先割合別処理日数

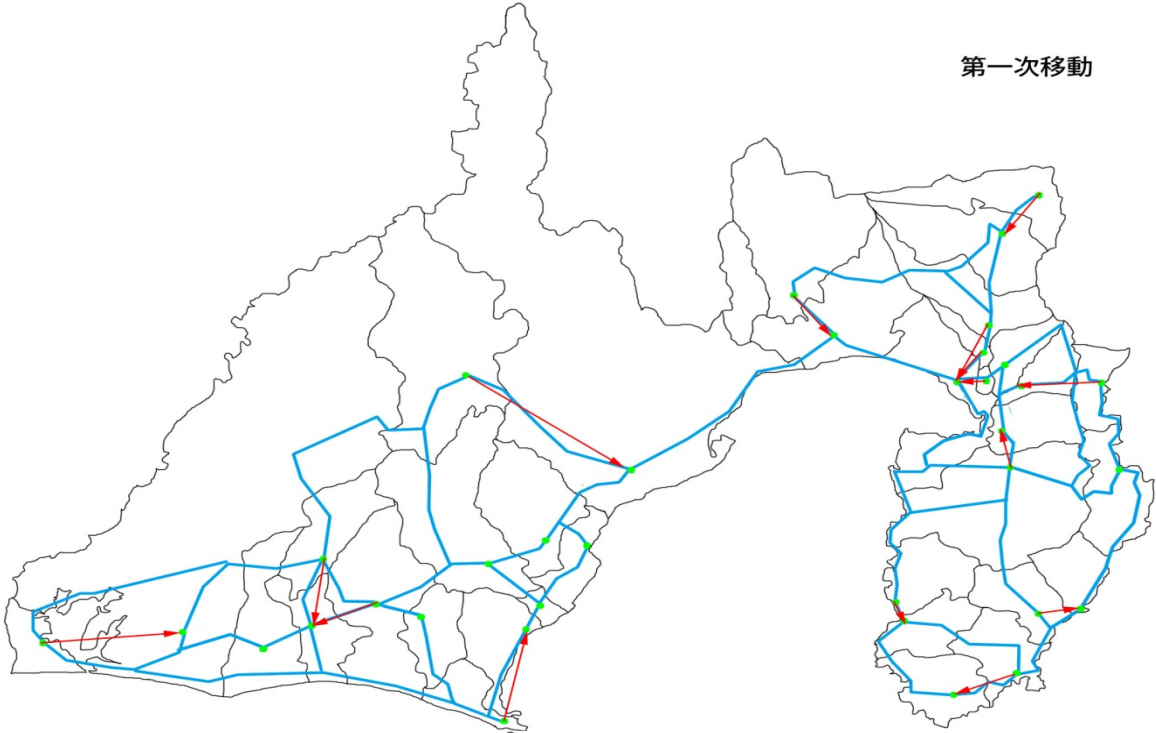
市町名	単独	全域連携処理日数 優先派遣割合別					
	処理日数	優先割合0%	優先割合 10%	優先割合20%	優先割合 30%	優先割合 40%	優先割合 50%
湖西市	9.92	9.92	10.56	10.86	14.01	14.98	15.43
浜松市	13.28	13.04	14.33	14.47	15.19	15.25	15.38
磐田市	30.09	14.47	14.38	13.98	7.00	5.50	4.81
袋井市	36.87	14.89	14.97	14.77	9.76	7.68	6.47
森町	13.84	13.63	15.00	13.72	11.45	11.84	12.77
掛川市	9.80	9.57	10.74	10.83	10.95	11.25	12.99
菊川市	23.32	13.87	11.31	10.37	5.75	4.37	3.51
御前崎市	4.28	4.28	4.68	4.63	6.15	7.32	9.13
牧之原市	5.85	5.20	5.72	5.87	7.28	8.61	10.86
吉田町	10.60	5.95	6.55	6.73	8.35	9.89	12.21
島田市	4.77	4.29	4.88	4.75	6.08	7.28	8.31
藤枝市	8.78	6.26	7.03	6.84	8.82	10.55	12.58
焼津市	14.08	7.47	8.29	8.27	10.48	12.48	14.03
静岡市	36.43	14.64	14.93	15.00	14.88	14.59	13.71
川根本町	1.24	1.24	1.37	1.36	1.73	2.12	2.64
富士宮市	11.50	11.49	12.66	12.63	13.68	14.35	14.28
富士市	16.82	12.71	12.92	12.90	13.40	13.85	14.35
御殿場市	9.15	8.33	9.95	9.30	11.43	13.94	13.33
小山町	6.76	6.78	8.01	7.36	9.33	10.49	13.01
裾野市	5.07	4.96	5.51	5.21	6.96	7.94	10.08
長泉町	3.74	3.73	4.44	4.05	5.30	6.49	8.13
沼津市	43.77	12.21	11.96	12.02	11.66	11.45	10.99
清水町	21.27	5.21	5.67	5.84	7.19	8.21	10.14
三島市	126.10	10.50	10.28	10.88	10.95	10.68	10.10
函南町	22.09	6.77	6.30	6.14	2.71	2.10	1.63
熱海市	4.94	4.93	5.82	5.34	6.88	8.00	10.07
伊豆の国市	7.14	4.13	4.48	4.63	5.99	6.82	8.86
伊豆市	1.59	1.58	1.75	1.78	2.37	2.62	3.37
伊東市	7.36	5.67	6.30	6.19	7.86	9.34	11.15
西伊豆町	2.94	2.95	3.33	3.24	4.39	4.87	6.35
松崎町	5.34	3.89	4.36	4.25	5.58	6.14	8.07
河津町	1.44	1.44	1.60	1.61	2.18	2.51	2.78
東伊豆町	3.74	2.18	2.35	2.43	3.05	3.72	4.11
下田市	11.97	5.16	5.68	5.67	7.21	8.33	10.06
南伊豆町	4.53	4.34	5.09	5.00	6.08	7.83	8.23
最大処理日数	14.80	14.89	15.00	15.00	15.19	15.25	15.43

優先市町名	優先割合0%	優先割合 10%	優先割合20%	優先割合 30%	優先割合 40%	優先割合 50%
磐田市	14.47	14.38	13.98	7.00	5.50	4.81
袋井市	14.89	14.97	14.77	9.76	7.68	6.47
菊川市	13.87	11.31	10.37	5.75	4.37	3.51
静岡市	14.64	14.93	15.00	14.88	14.59	13.71
沼津市	12.21	11.96	12.02	11.66	11.45	10.99
三島市	10.50	10.28	10.88	10.95	10.68	10.10
函南町	6.77	6.30	6.14	2.71	2.10	1.63
全域	14.89	15.00	15.00	15.19	15.25	15.43

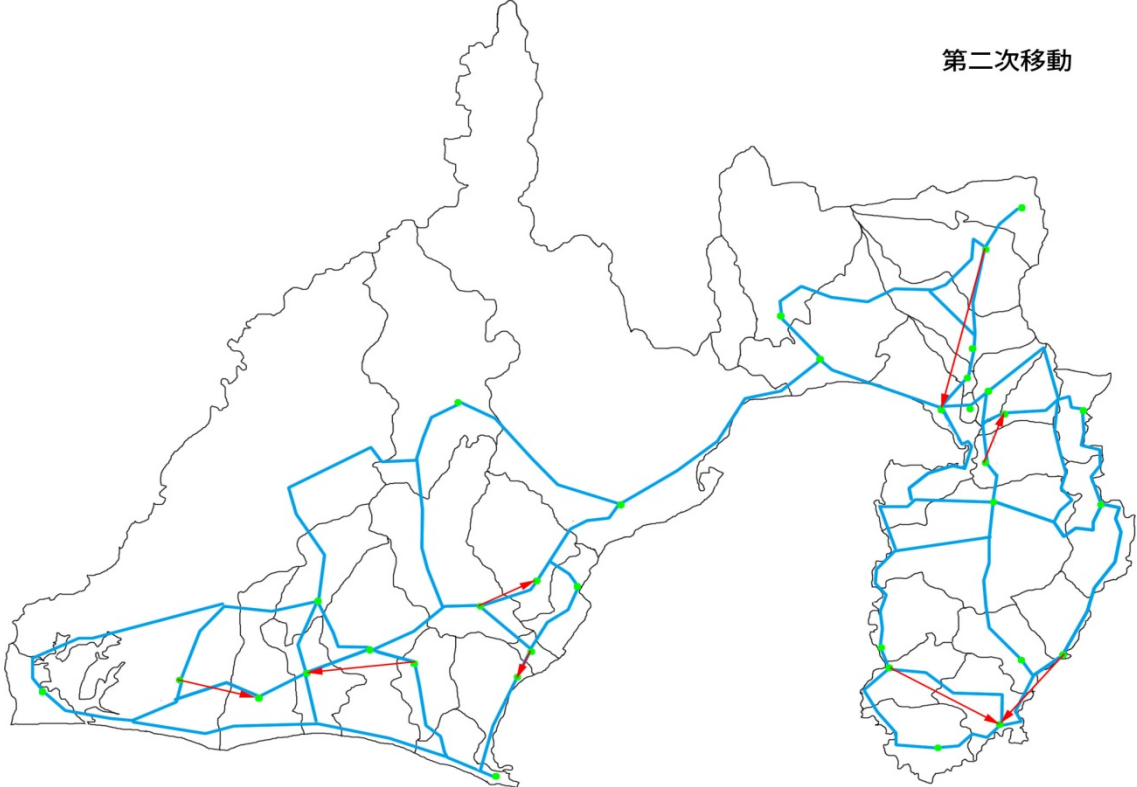
表-8 各モデルにおける処理日数

市町名	単独処理日数 a)	地域連携処理日数 b)	分断地域内連携処理日数 c)	全域連携処理日数 d)
湖西市	9.92	9.96	9.92	9.92
浜松市	13.28	13.04	12.44	13.04
磐田市	30.09	14.33	14.35	14.47
袋井市	36.87	14.03	15.58	14.89
森町	13.84	13.74	13.57	13.63
掛川市	9.80	9.50	9.78	9.57
菊川市	23.32	9.27	15.04	13.87
御前崎市	4.28	4.28	4.21	4.28
牧之原市	5.85	5.86	5.21	5.20
吉田町	10.60	6.72	5.97	5.95
島田市	4.77	4.64	4.49	4.29
藤枝市	8.78	6.53	6.35	6.26
焼津市	14.08	8.16	7.53	7.47
静岡市	36.43	17.01	15.92	14.64
川根本町	1.24	1.24	1.24	1.24
富士宮市	11.50	11.50	11.66	11.49
富士市	16.82	14.33	12.71	12.71
御殿場市	9.15	7.58	8.10	8.33
小山町	6.76	6.78	6.77	6.78
裾野市	5.07	4.59	5.01	4.96
長泉町	3.74	3.73	3.73	3.73
沼津市	43.77	16.96	12.21	12.21
清水町	21.27	17.14	5.23	5.21
三島市	126.10	9.28	10.83	10.50
函南町	22.09	5.44	6.79	6.77
熱海市	4.94	4.93	4.73	4.93
伊豆の国市	7.14	4.30	4.13	4.13
伊豆市	1.59	1.58	1.58	1.58
伊東市	7.36	6.82	5.66	5.67
西伊豆町	2.94	2.95	2.95	2.95
松崎町	5.34	3.89	3.89	3.89
河津町	1.44	1.45	1.45	1.44
東伊豆町	3.74	2.18	2.18	2.18
下田市	11.97	5.27	5.16	5.16
南伊豆町	4.53	4.11	4.34	4.34
最大処理日数	126.10	17.14	15.92	14.89

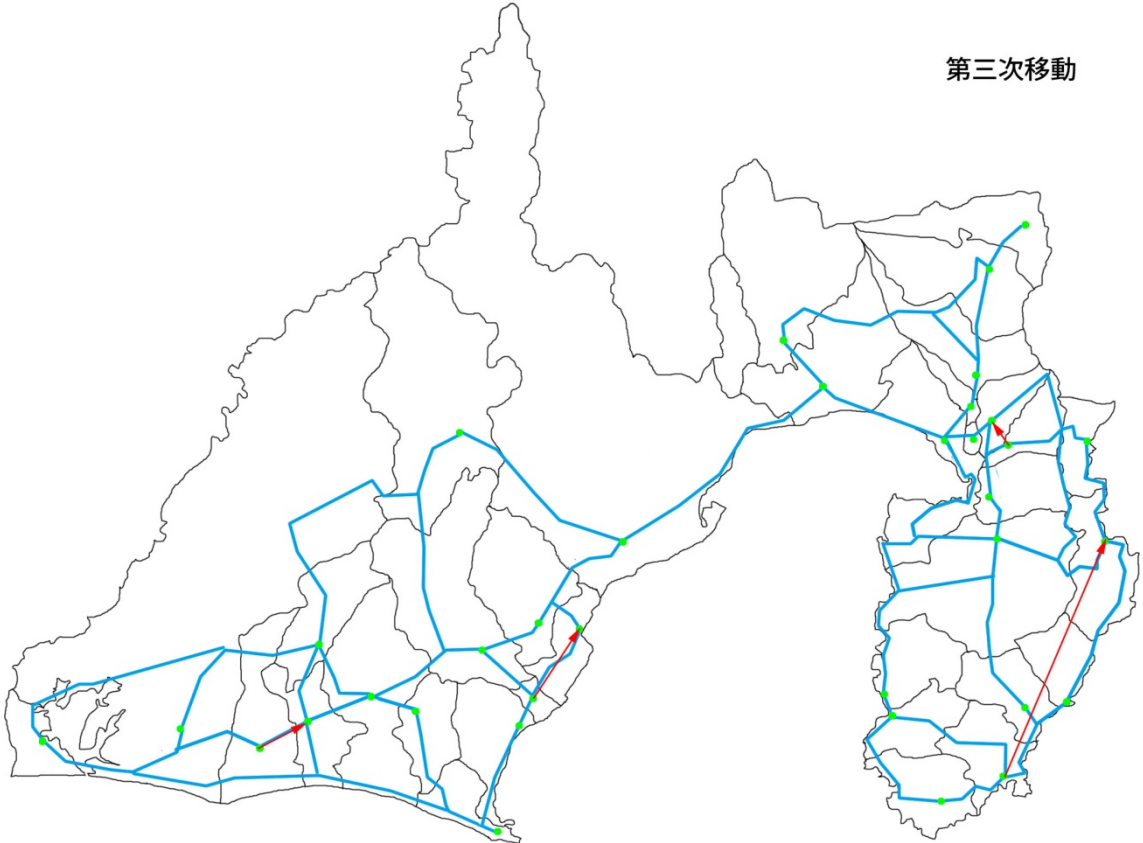
第一次移動



第二次移動



第三次移動



第4～7次移動

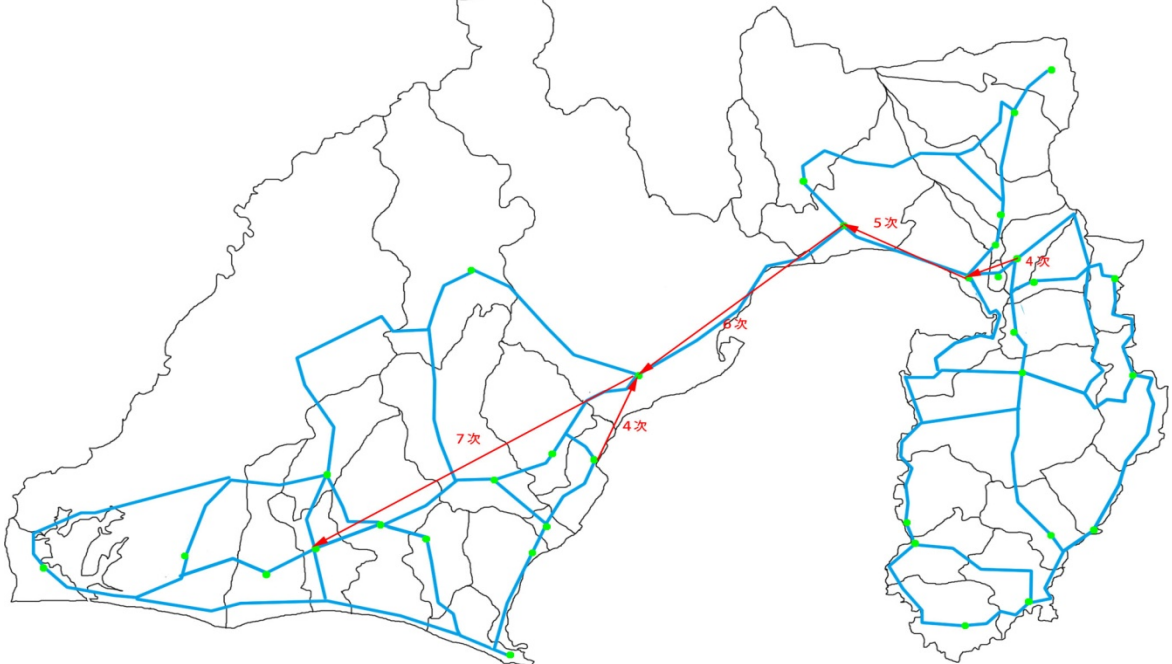
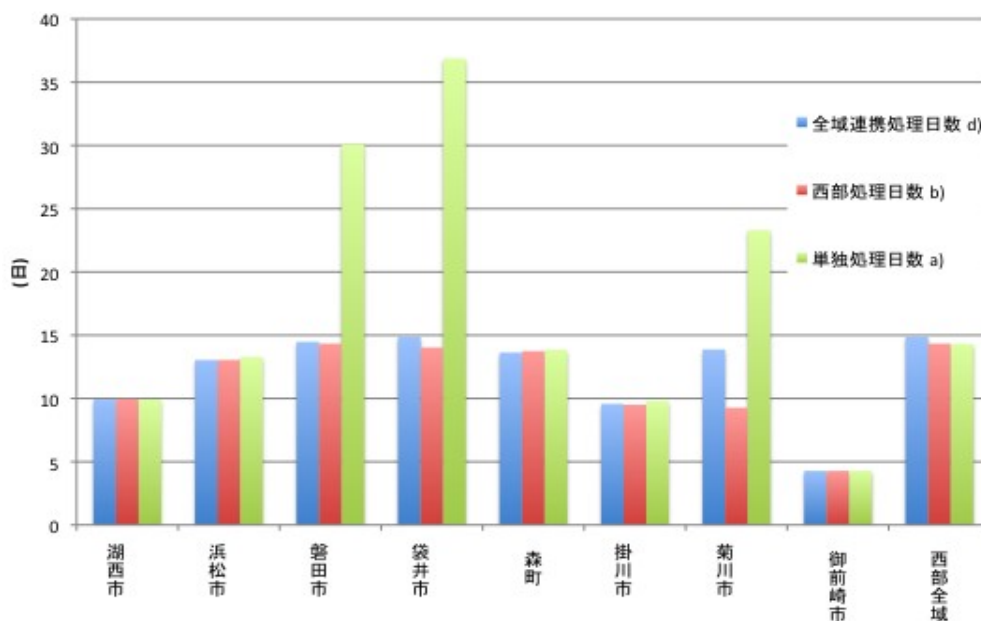


図-12 全域連携モデルにおける重機移動歴

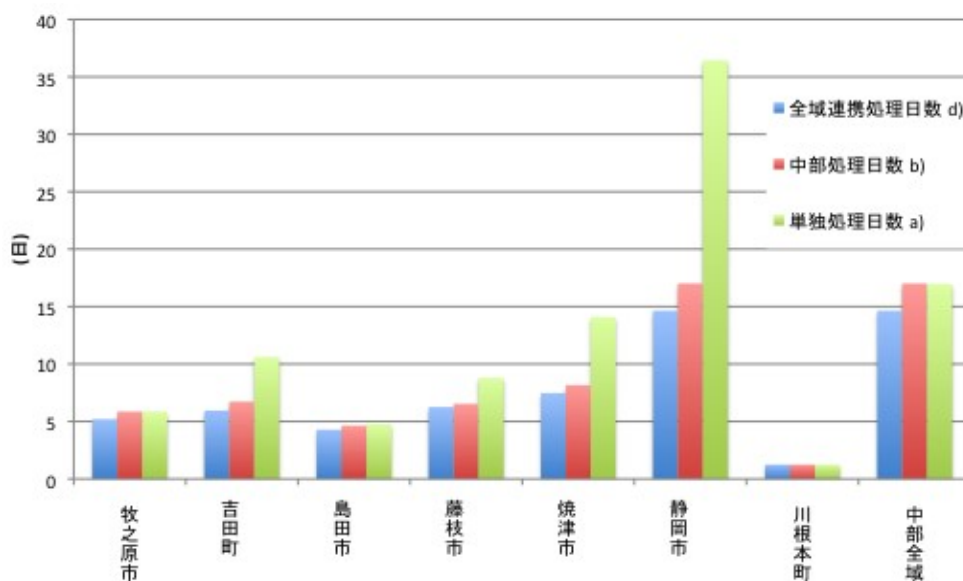


#### 4.2 静岡県内4地域内での連携を考慮したモデル

静岡県を西部，中部，東部，伊豆の4地域に分割し，その域内のみで建設業者が連携してがれき処理を行う場合をシミュレートした．これより算出される市町ごとの処理日数（以後，地域連携処理日数b）と呼ぶ．）と，全域処理日数を比較して，地域ごとに図-13a)～e)に示す．またその数値は表-8にすでに示した．中部地域と東部地域では全体の連携を行った場合より，地域内での連携での処理日数が上回り，反対に西部地域，伊豆地域では全体の連携を行った場合の処理日数に比べ，地域内での連携の処理日数が下回っている．それぞれの地域間においてのがれき量に対しての重機数の差，重機の偏在が影響していると考えられる．静岡県を4地域に分けた場合，2地域ごとに重機を派遣する側，重機を派遣される側に分かっていると，この結果より考えられる．

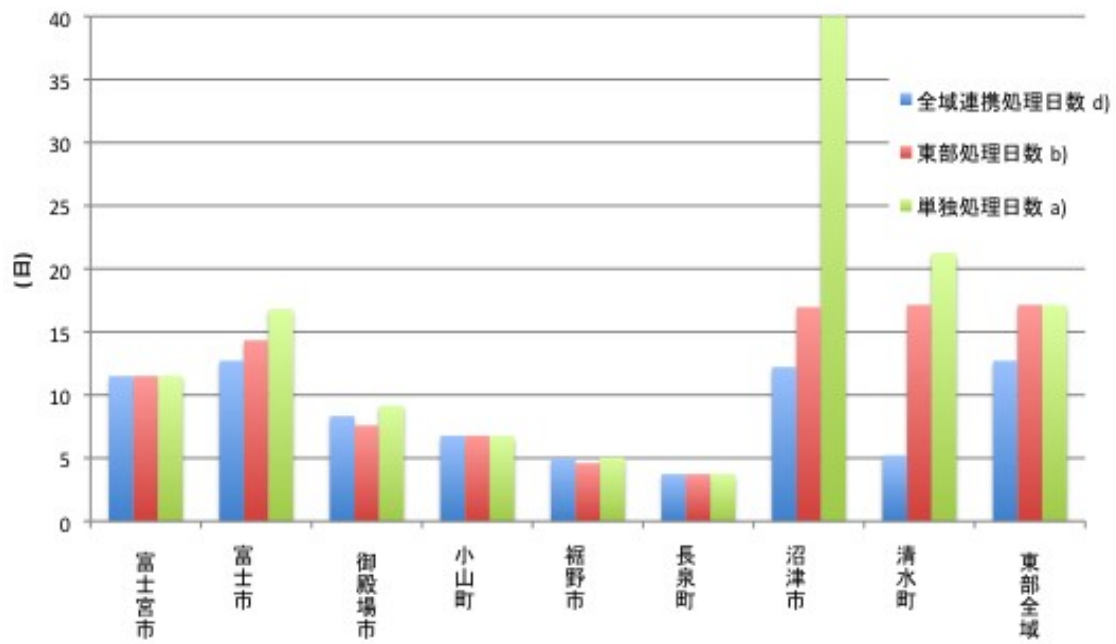


a) 西部地域での連携結果



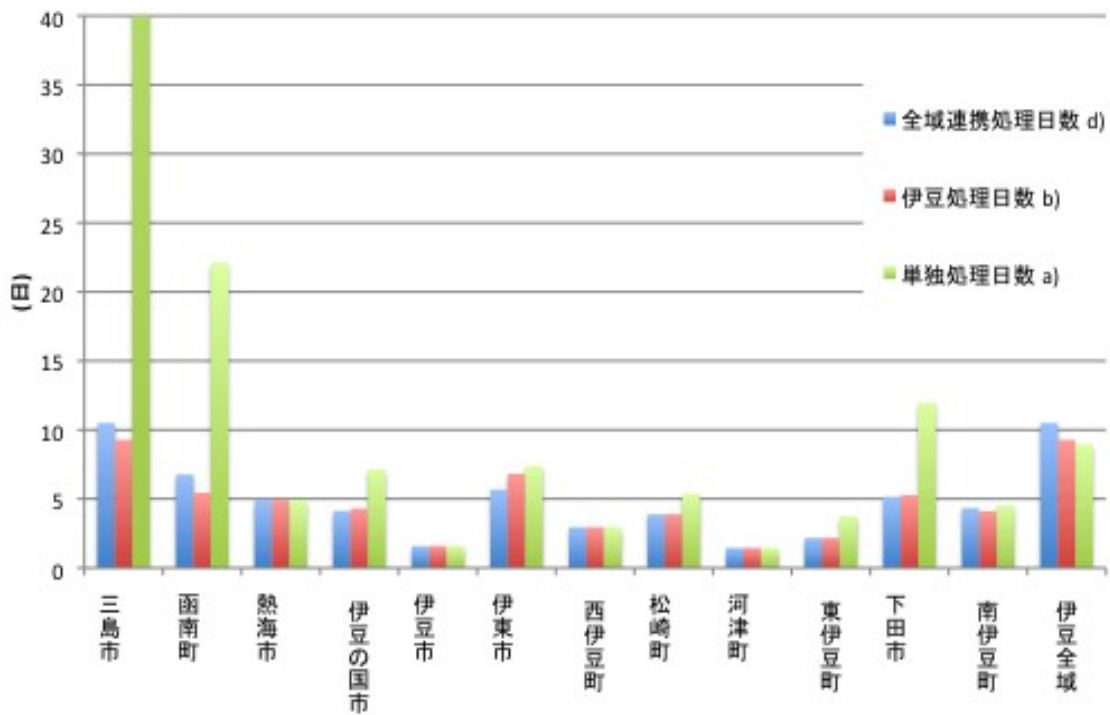
b) 中部地域での連携結果



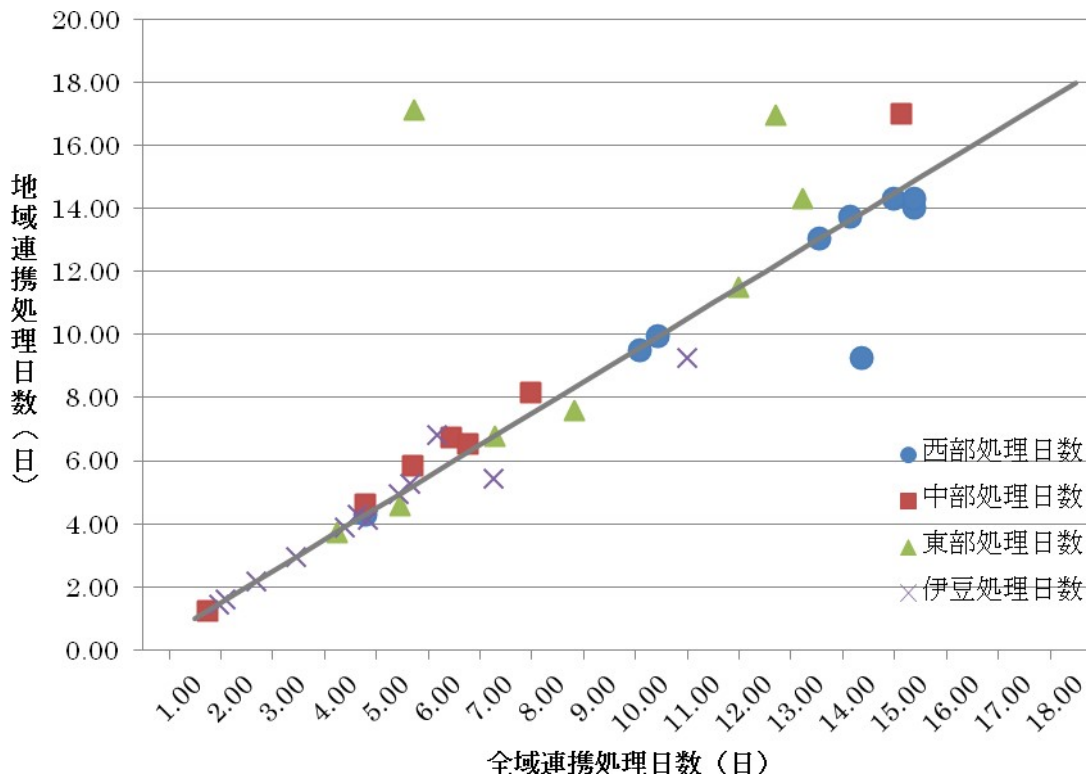


c) 東部地域の連携結果

c)



d) 伊豆地域の連携結果



e) 市町別, 全域連携と地域連携の比較  
 図-13 地域連携モデル結果

#### 4.3 緊急輸送路の被害を想定したモデル

第3次被害想定<sup>5)</sup>より, 緊急輸送路の被害が想定される富士川付近の断層の影響により, 近辺の緊急輸送路が寸断される可能性が大きい. そこで, 富士川付近の緊急輸送路が寸断され静岡県が東西の地域に分断した場合のシミュレーションを行った. これにより算出される市町ごとの処理日数(以後, 分断地域内連携処理日数と呼ぶ.)と, 全域処理日数を比較して, 西側地域における結果は図-14, 東側地域における結果は図-15に示す. 西側地域の大半の市町における分断地域内連携処理日数がわずかながら全域連携日数を上回る結果となった. 一方, 東側地域では分断地域内連携処理日数と全域連携日数との差がほぼない結果となった. この事より, 静岡県が東西に分断した場合, 全域での連携に対して西側の地域は, わずかながら重機作業効率が低下する事が判明した.

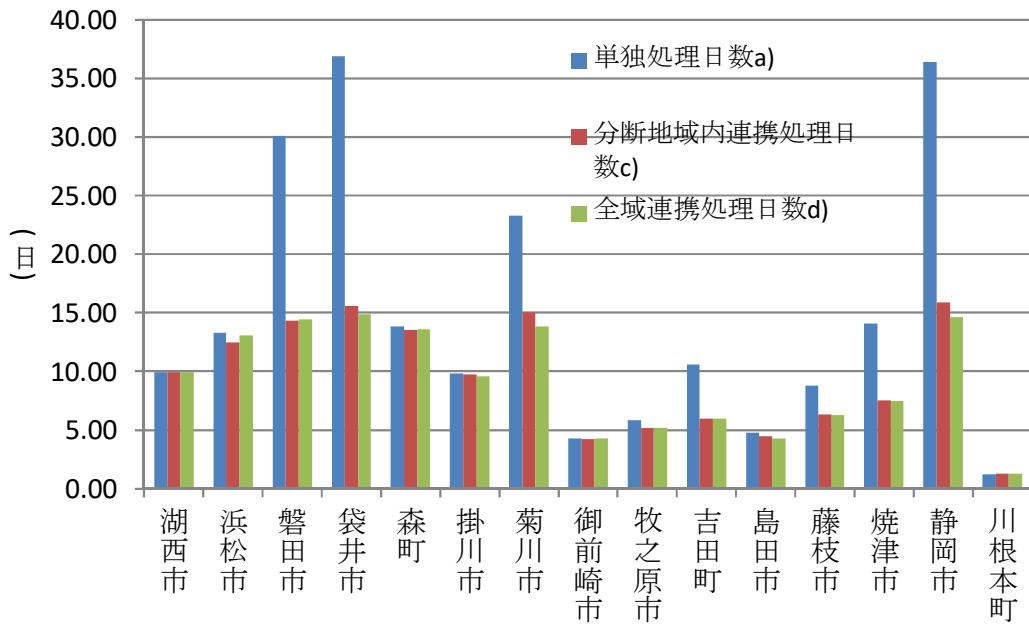


図-14 西側地域の連携結果

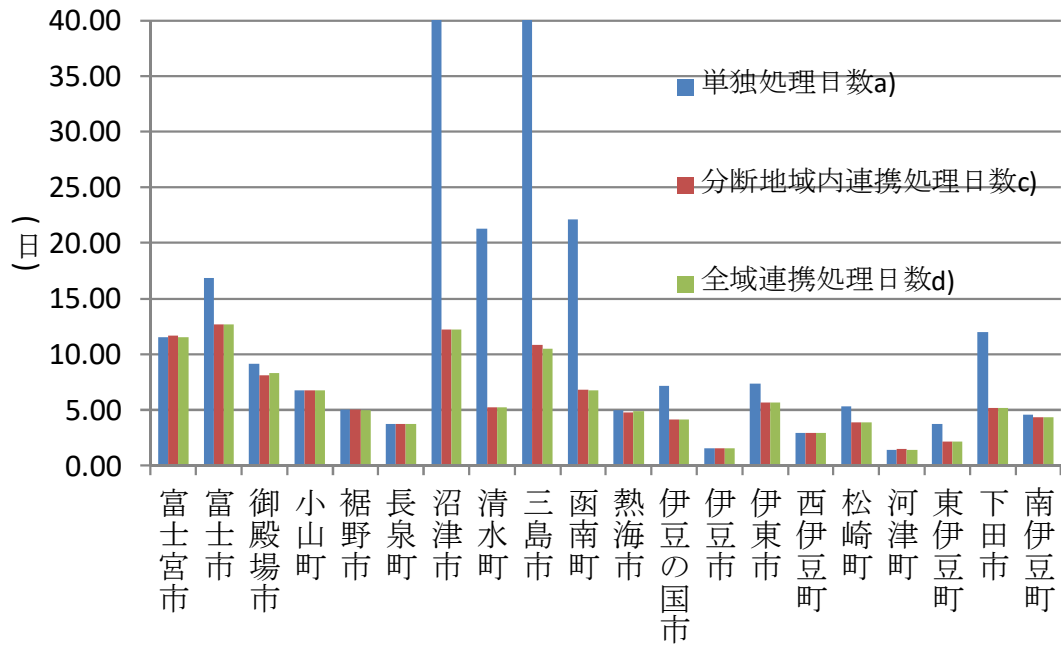


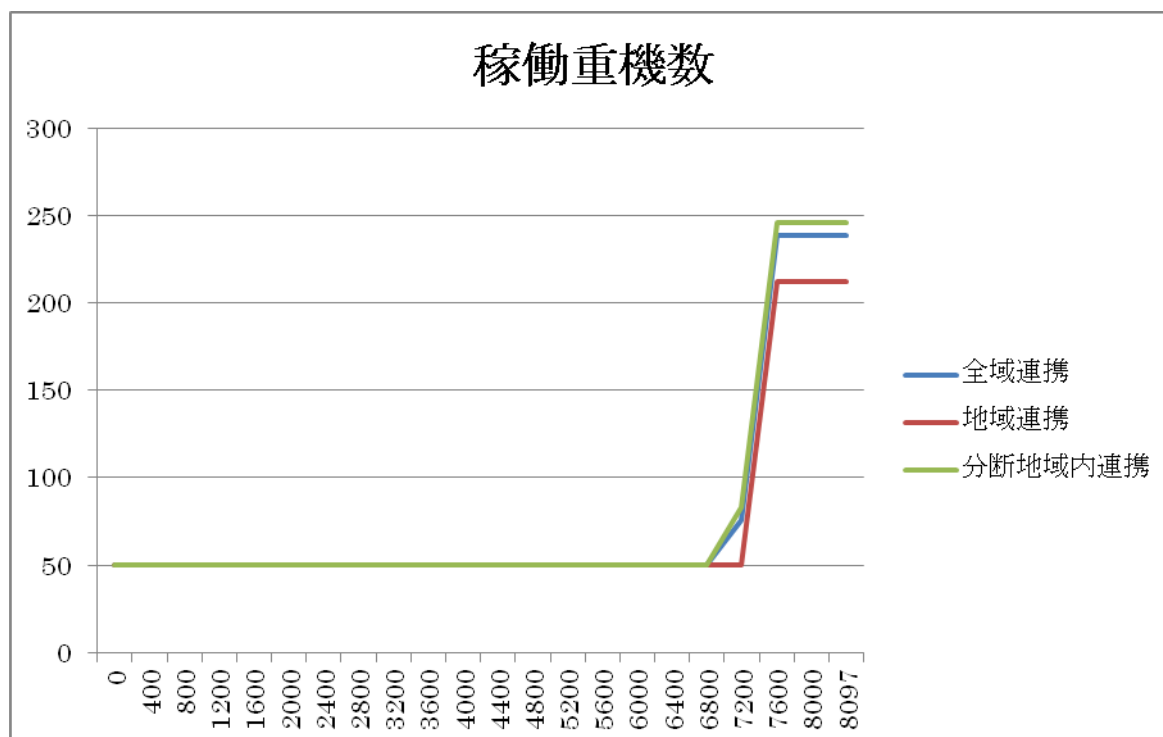
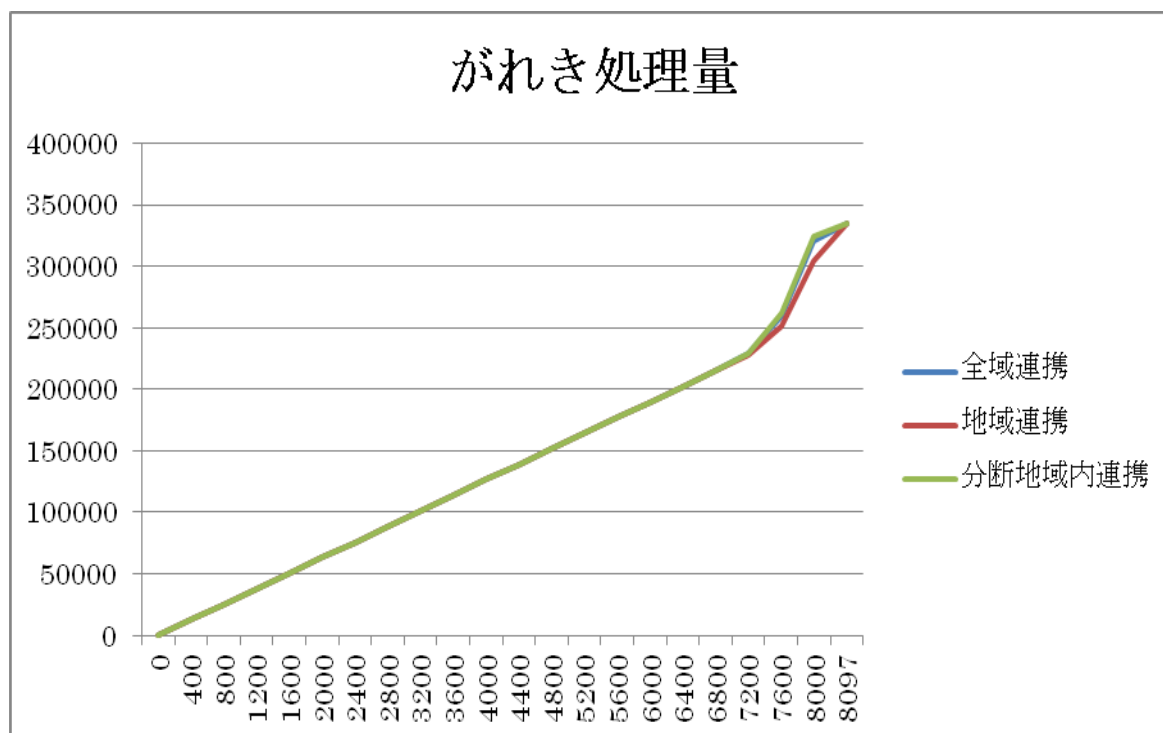
図-15 東側地域の連携結果

#### 4.4 各モデルにおける市町別がれき処理量と稼働重機数の推移

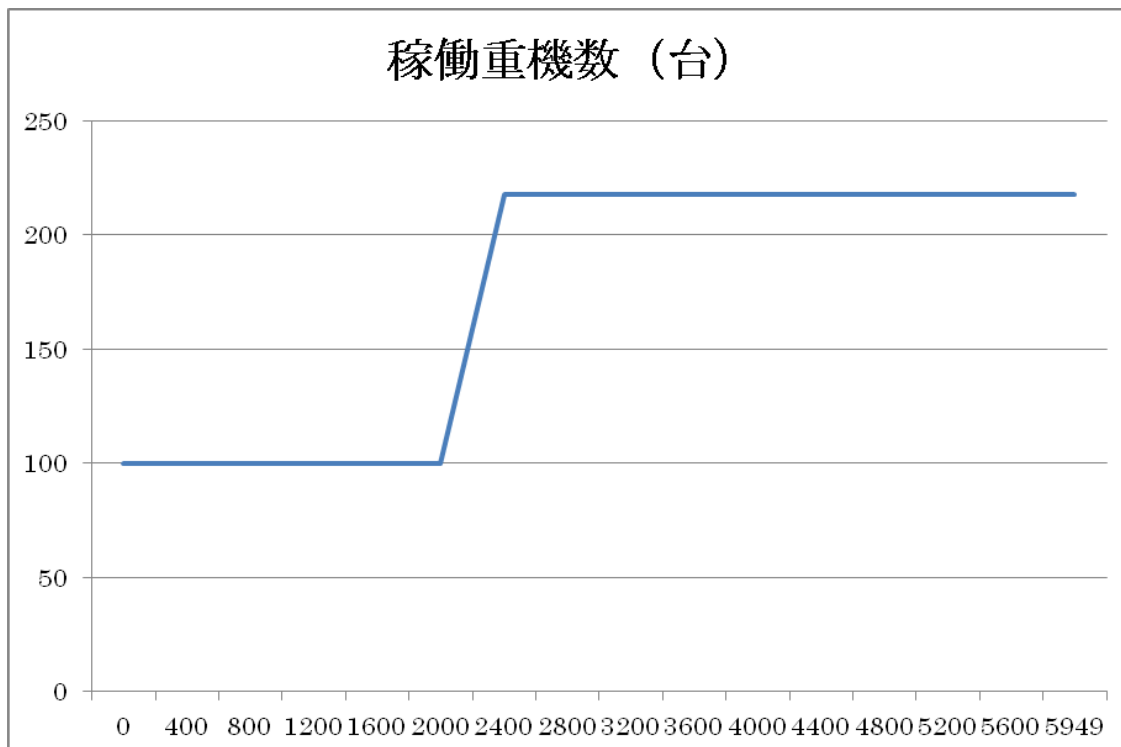
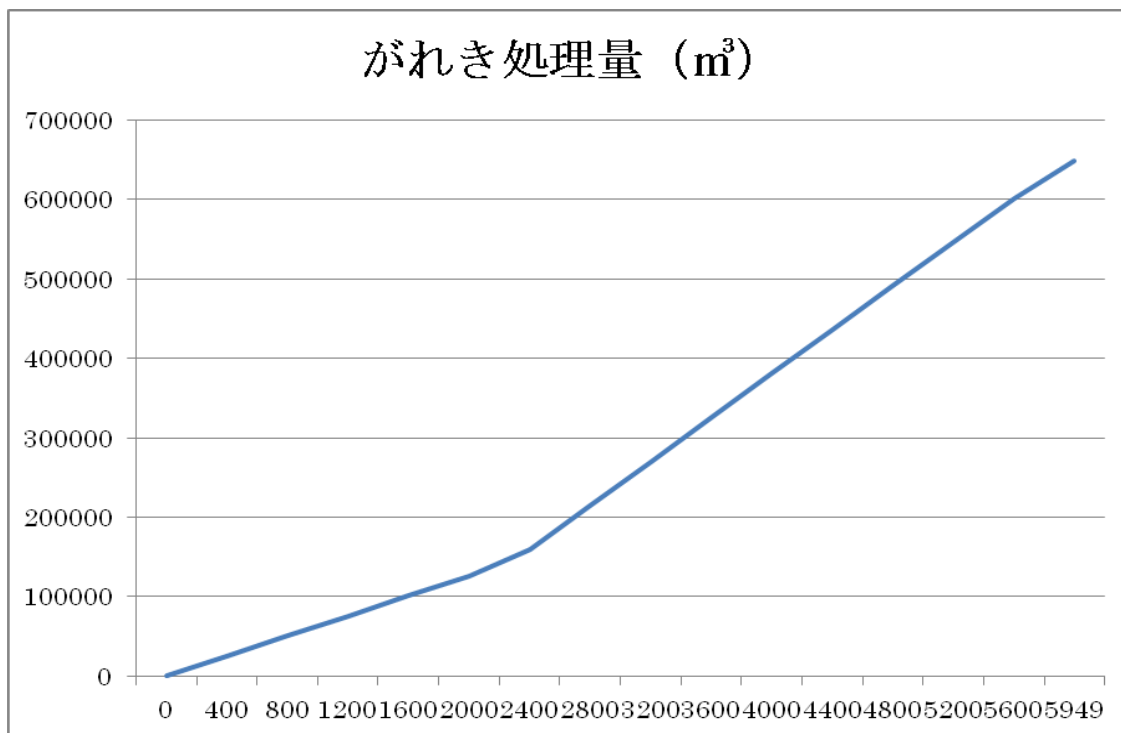
各モデルにおける市町別のがれき処理量、稼働重機数の時間変化を 400 分おきに測定し以下に全市町別がれき処理量, 稼働重機数を示す. シミュレーション全期間を 5 日間隔で前期, 中期, 後期の 3 段階に分けた場合. 中部, 東部, 伊豆地域は前期, 中期に多くの移動が見られ, 西部地域では後期に重機の移動が多いことが確認された.

連携範囲の違いによりがれき処理日数に変化があることが確認できた. 多くの市町で, 連携範囲の拡大に伴い, がれき処理日数が減少する傾向がある. 代表的な例は静岡市である. また一部市町では連携範囲に関わらず処理日数が変化しない結果となった. 例として伊豆の国市を挙げる. 処理日数が増加している市町も確認された. 例として御殿場市を挙げる. それぞれの要因は各市町間における発生がれき量と重機数の差であり, 重機の偏在が大きな影響を及ぼしていると考えられる.

# 伊東市

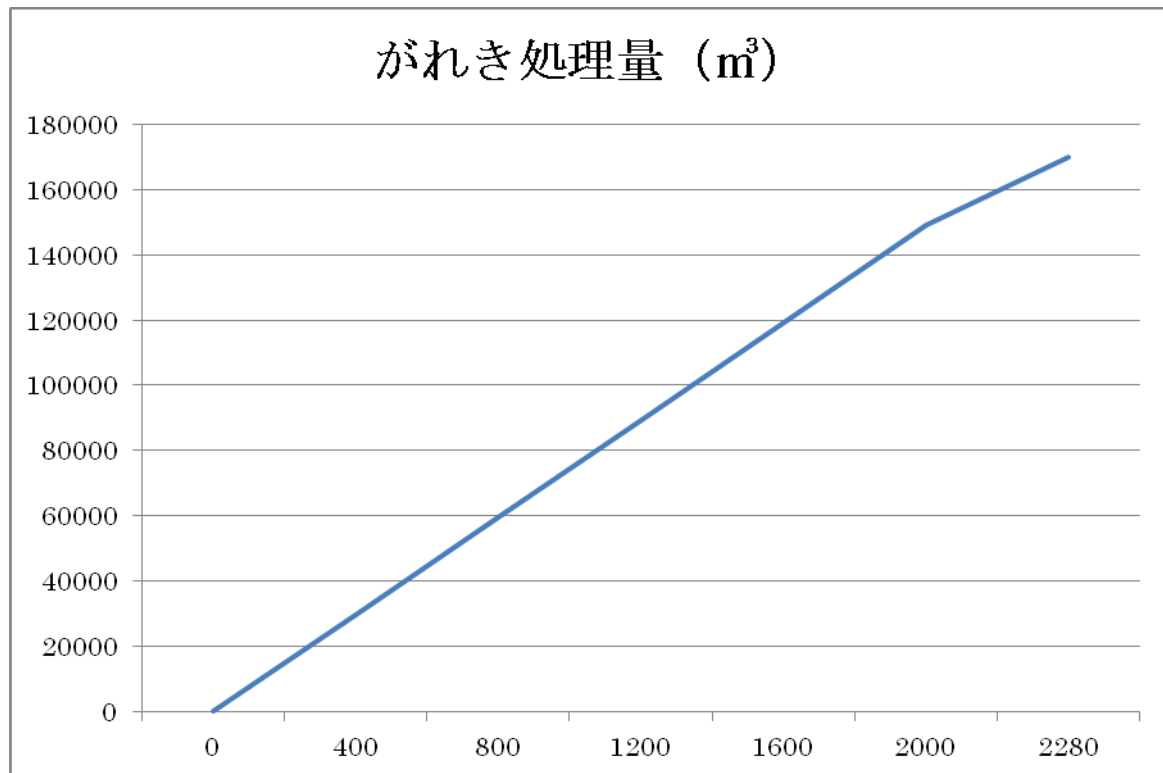


## 伊豆の国市



全てのモデルで同じ挙動を示した.

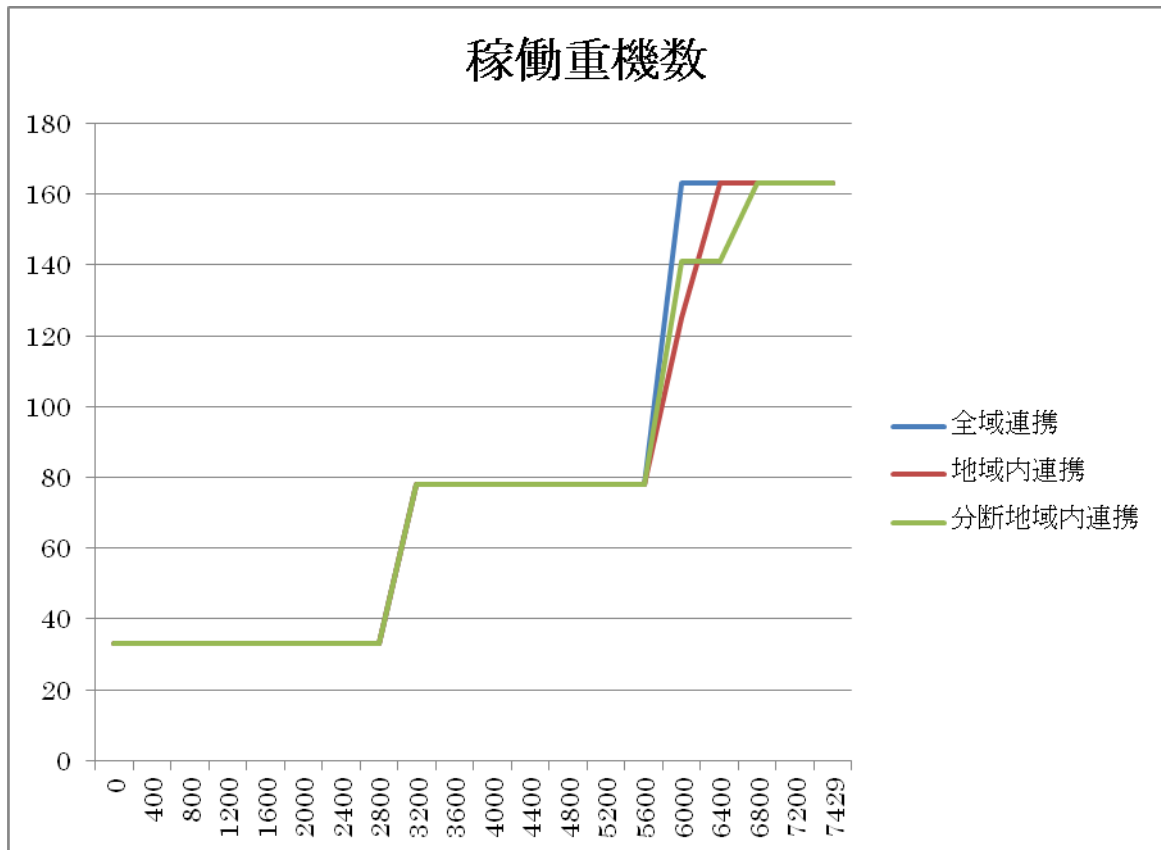
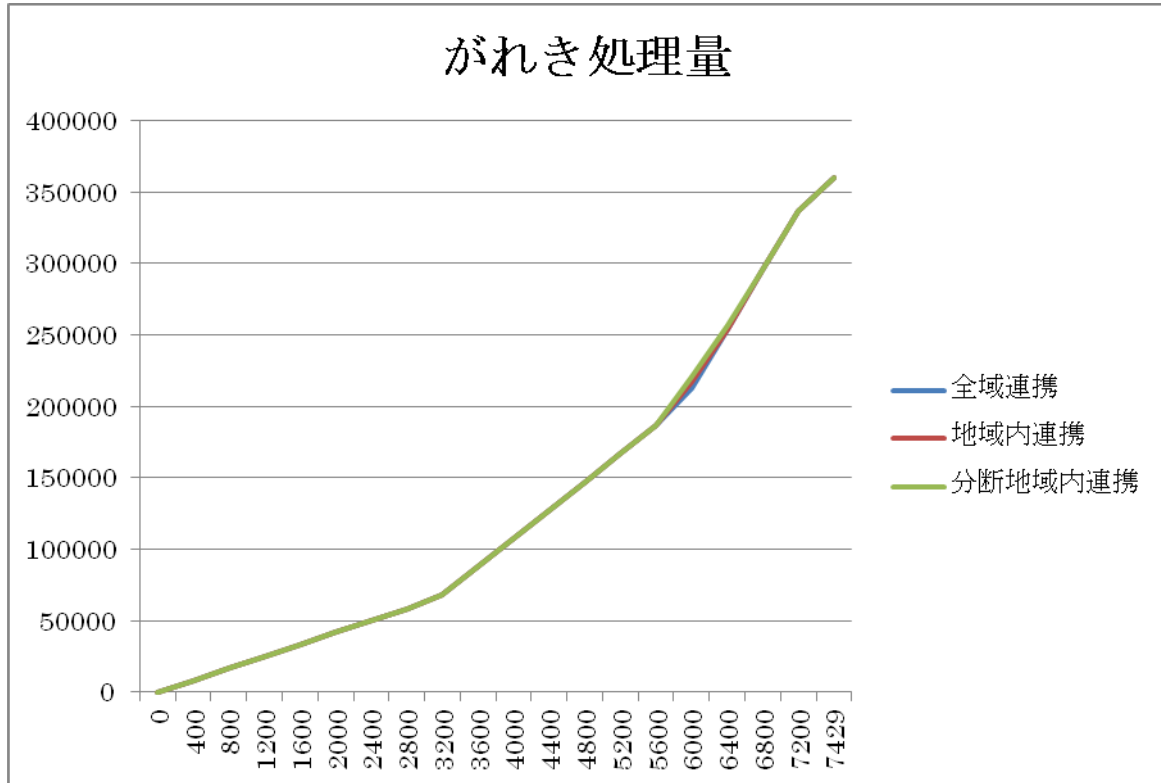
## 伊豆市



稼働重機数は 118 台で一定.

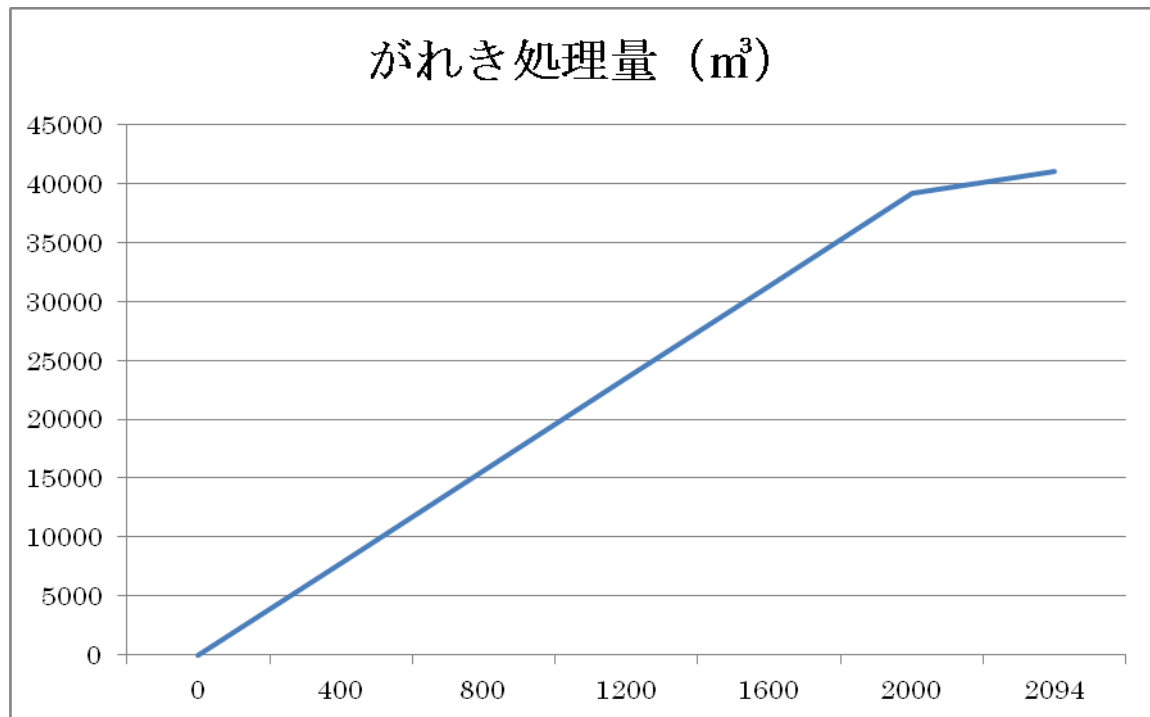
全てのモデルで同じ挙動を示した.

下田市



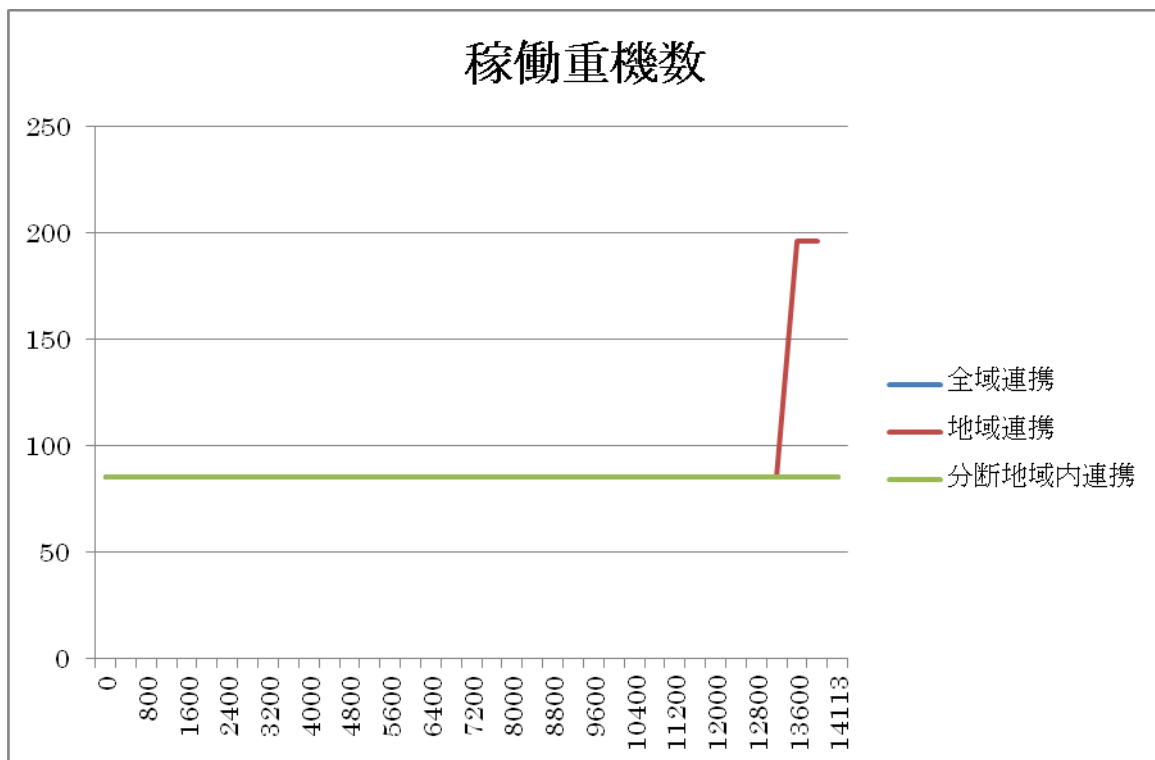
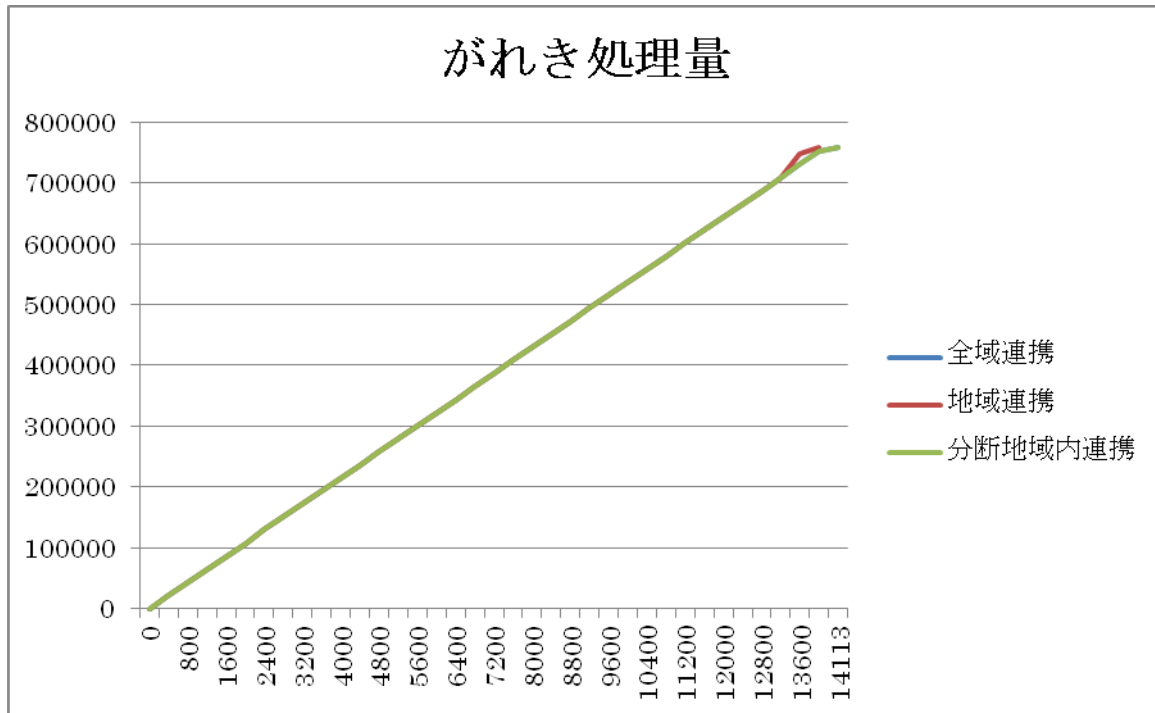


## 河津町

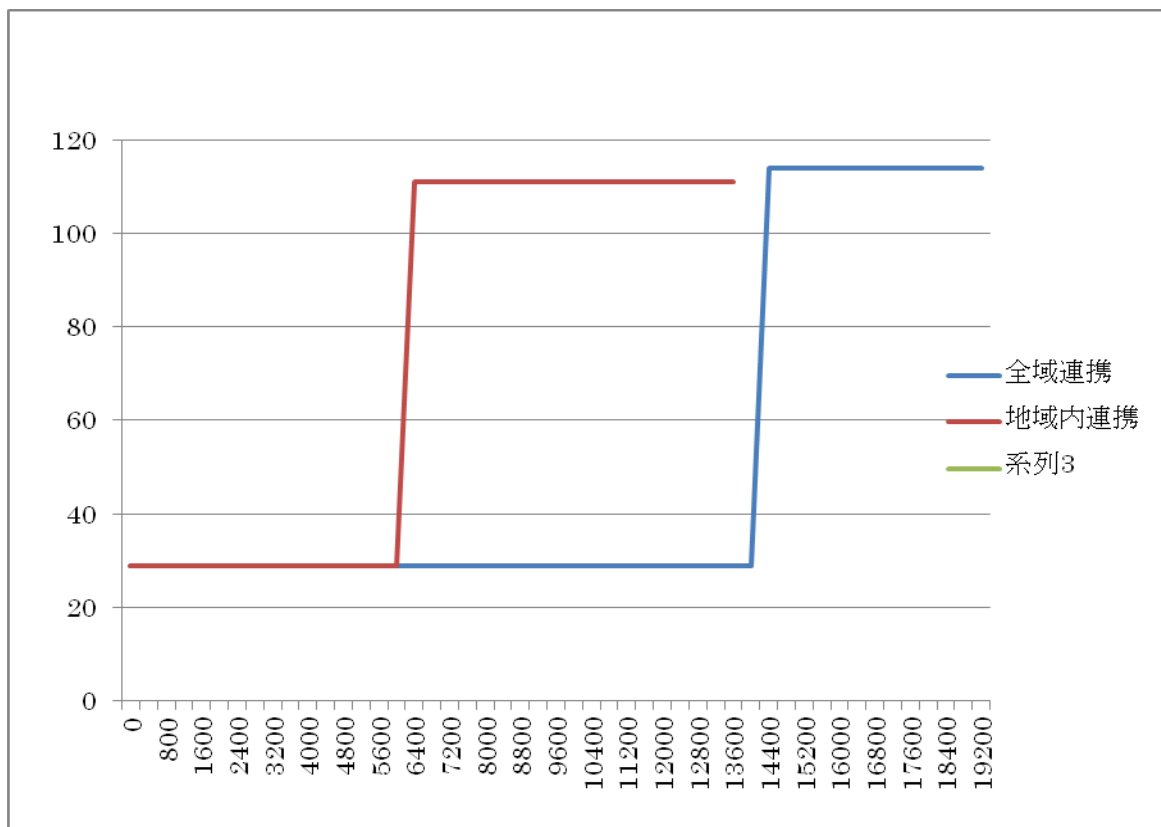
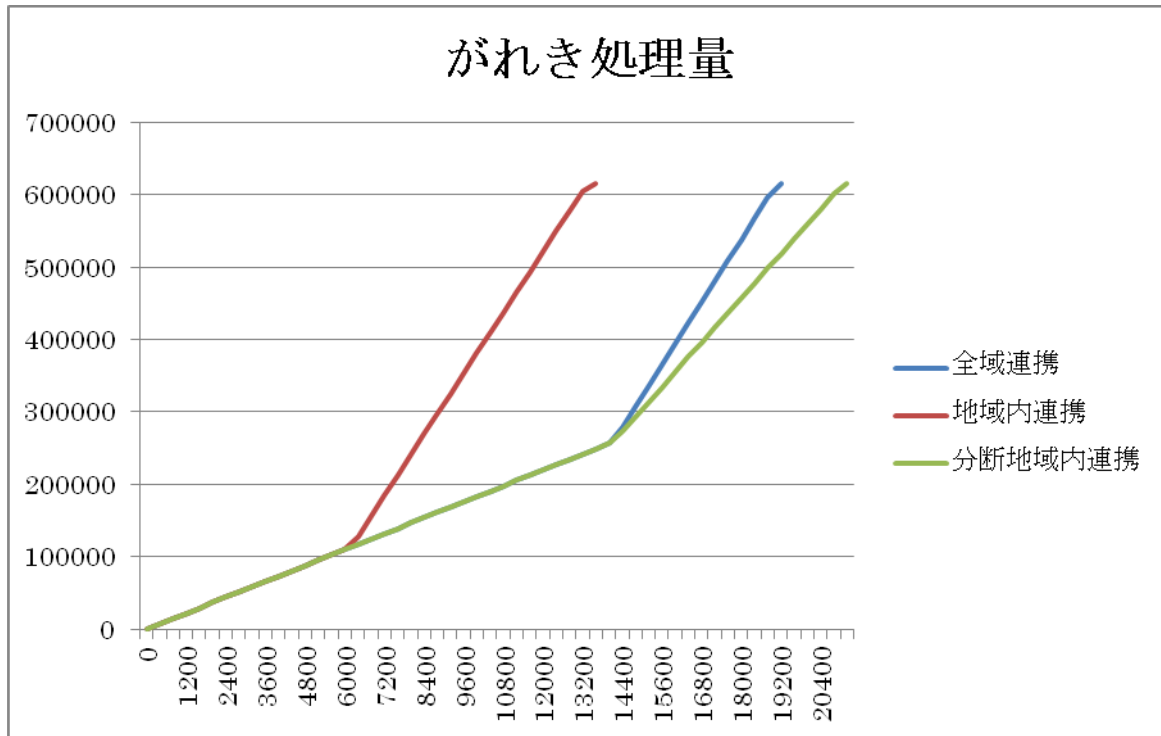


稼働重機数は 31 台で一定. また, 全てのモデルで同じ挙動を示した.

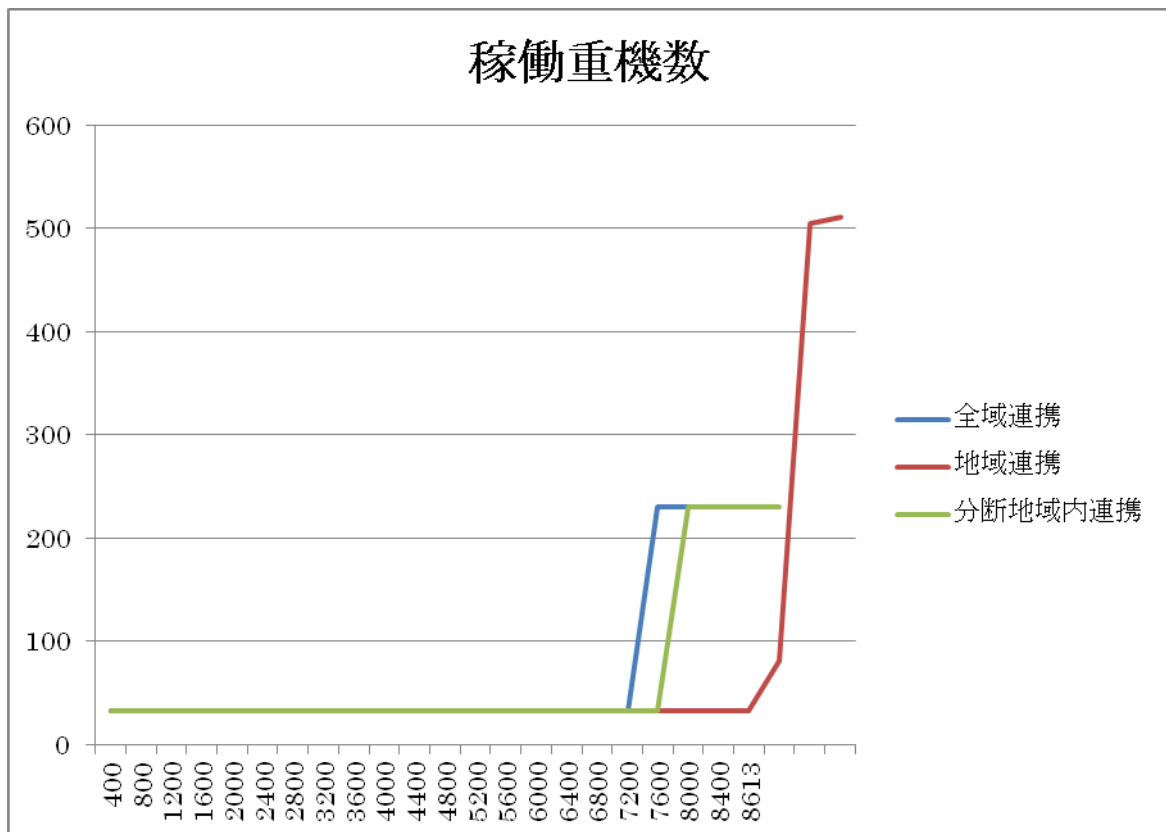
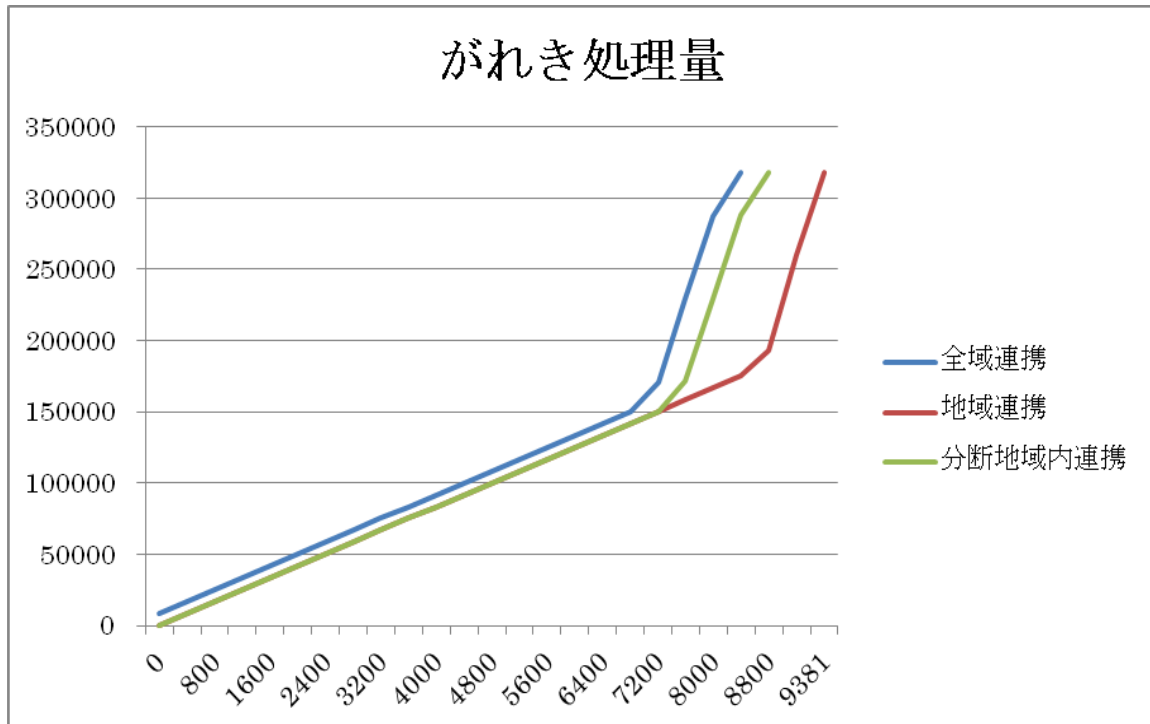
掛川市



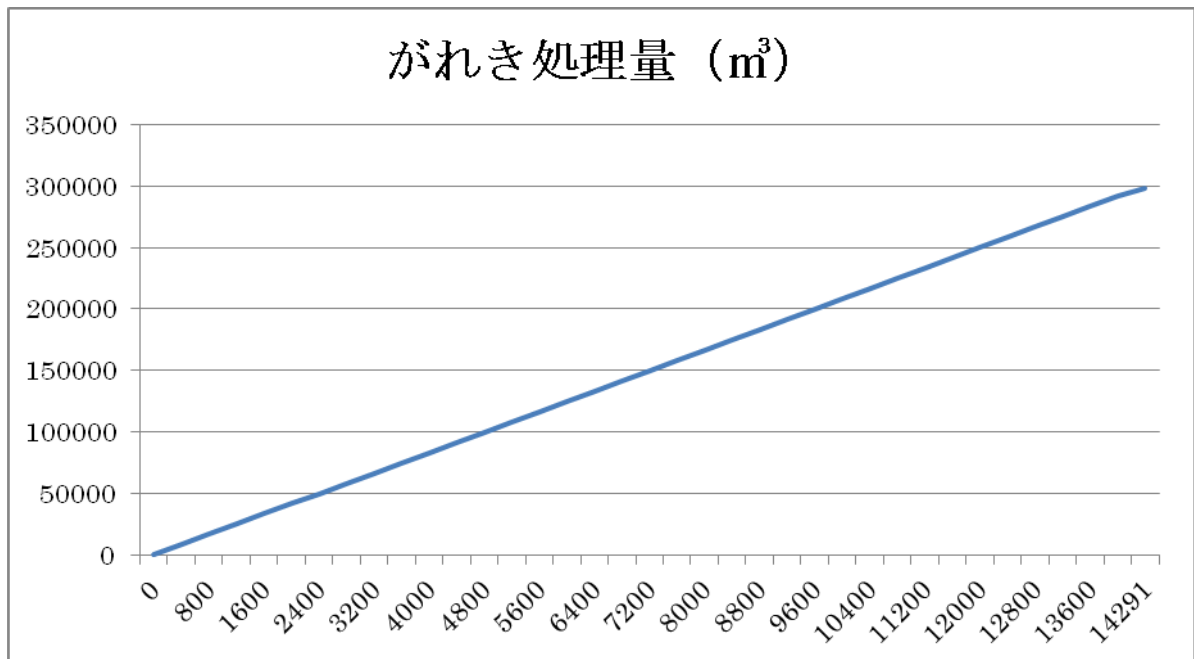
菊川市



# 吉田町

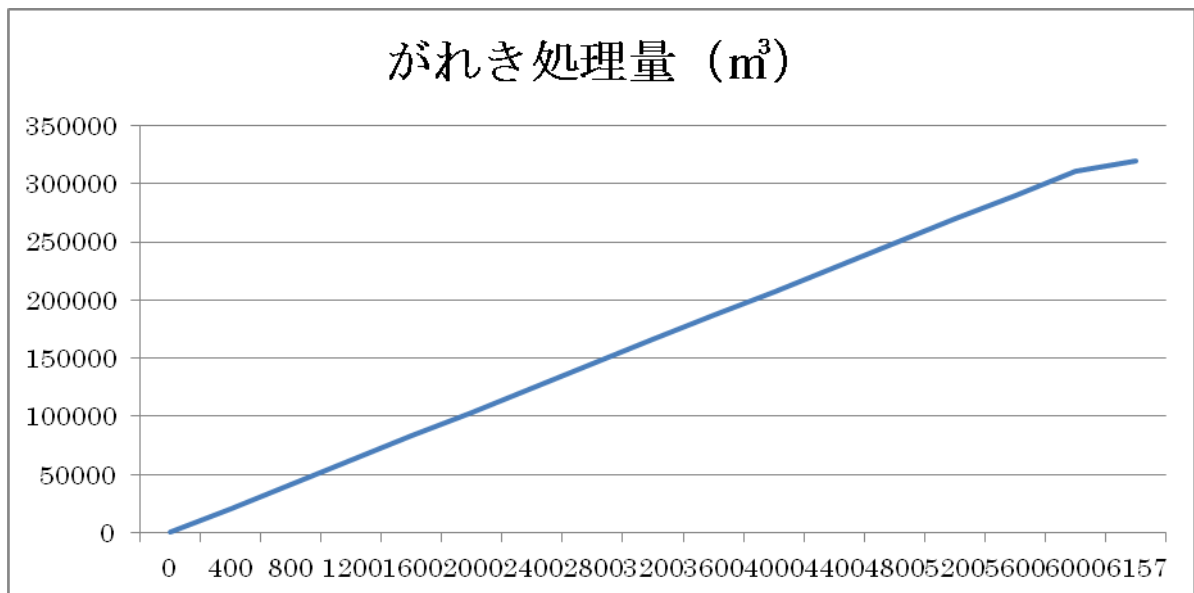


## 湖西市



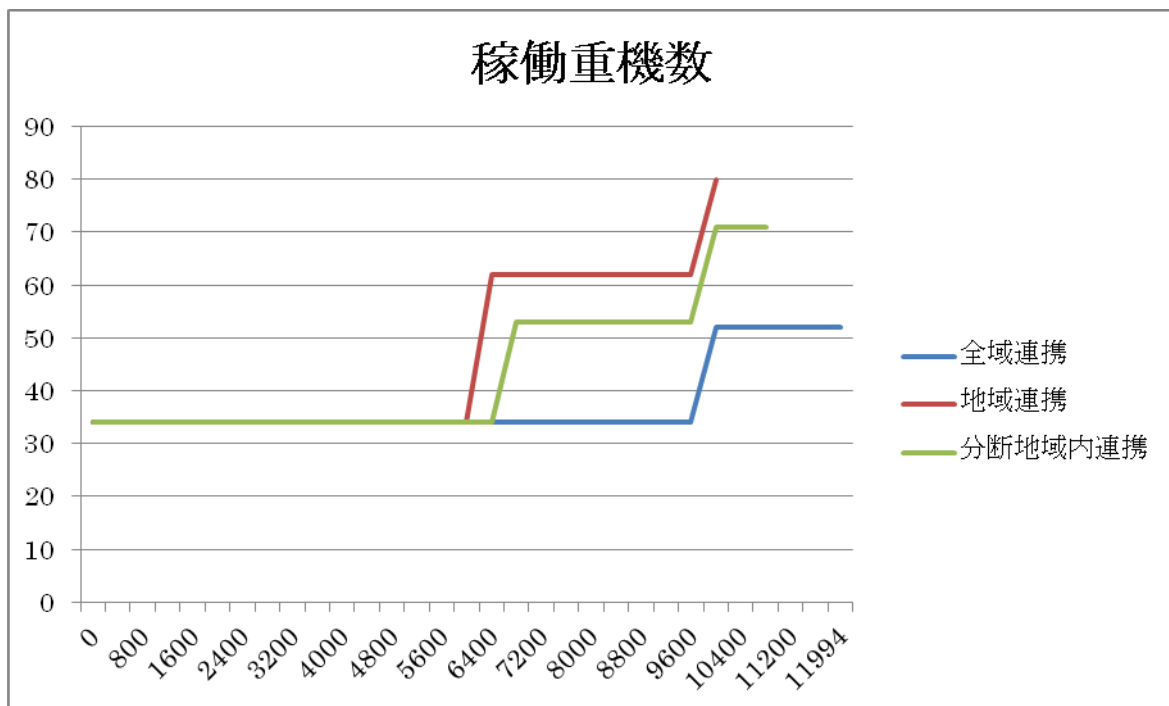
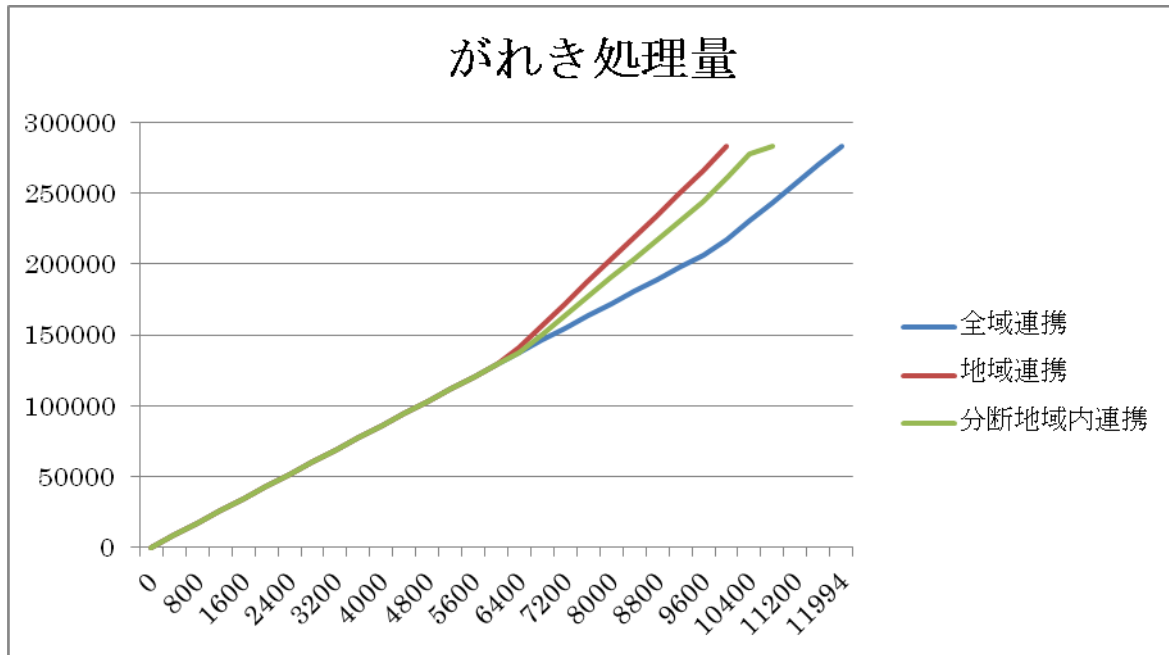
稼働重機数は 33 台で一定. 各モデル同じ挙動を示した.

## 御前崎

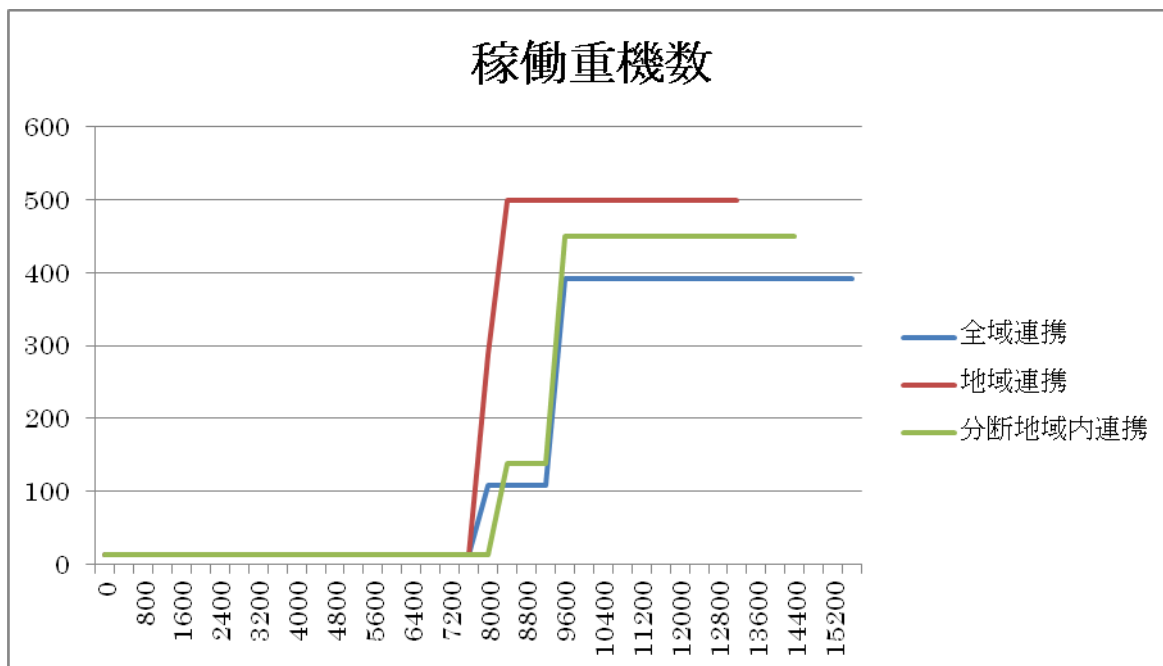
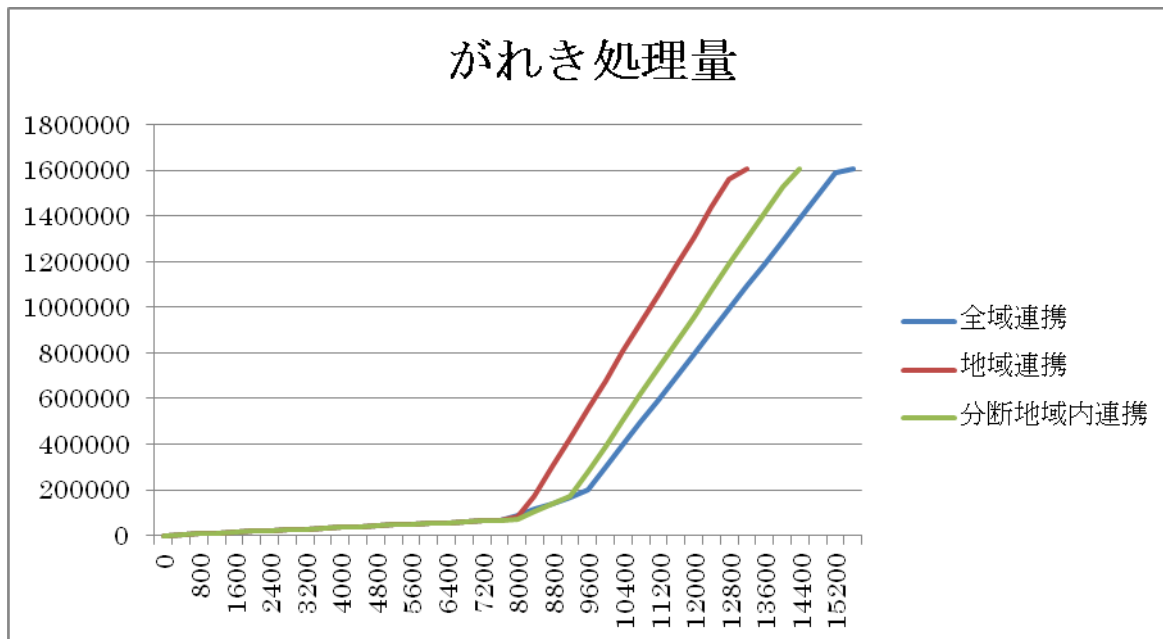


稼働重機数は 82 台で一定. 各モデル同じ挙動を示した.

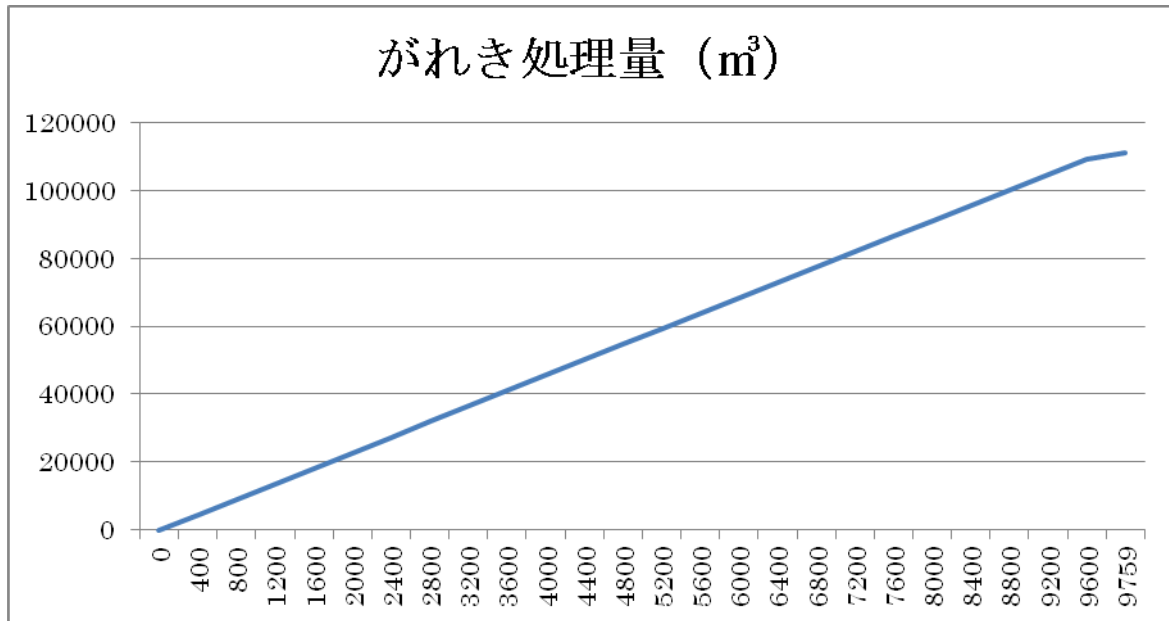
御殿場市



# 三島市



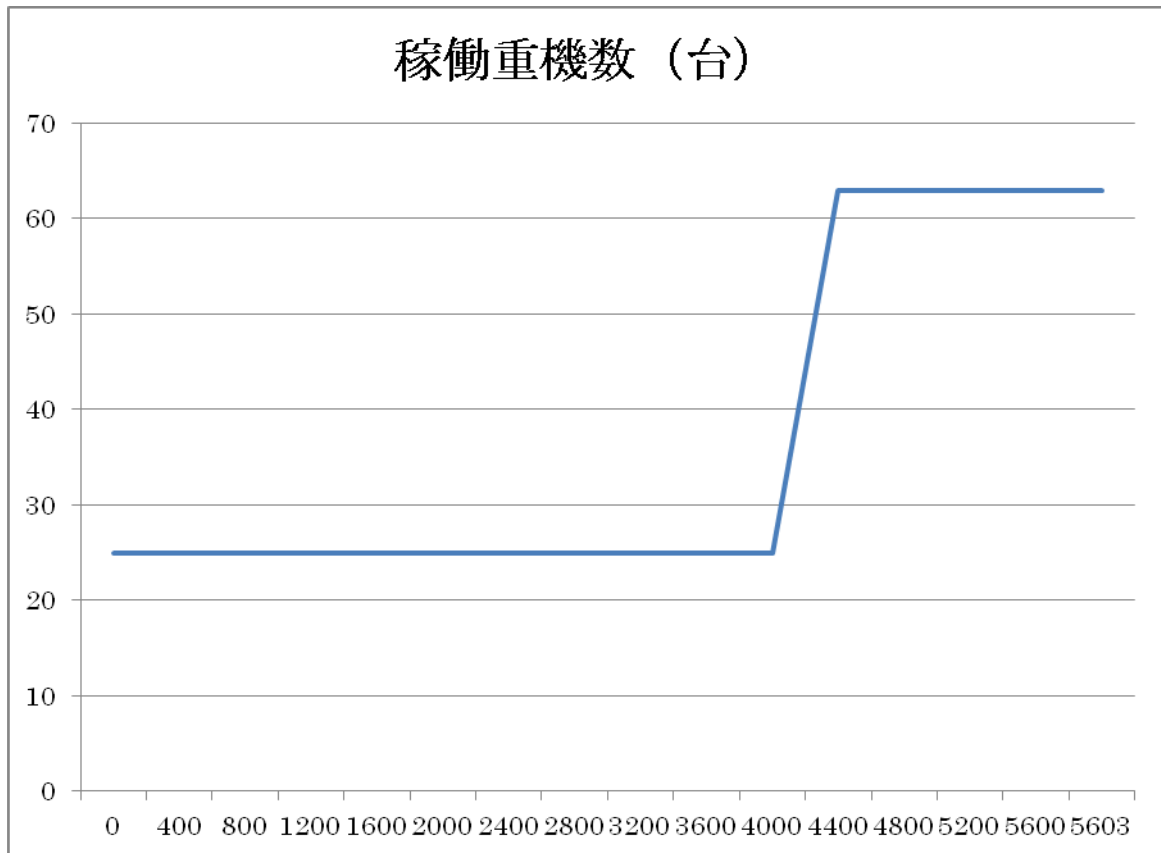
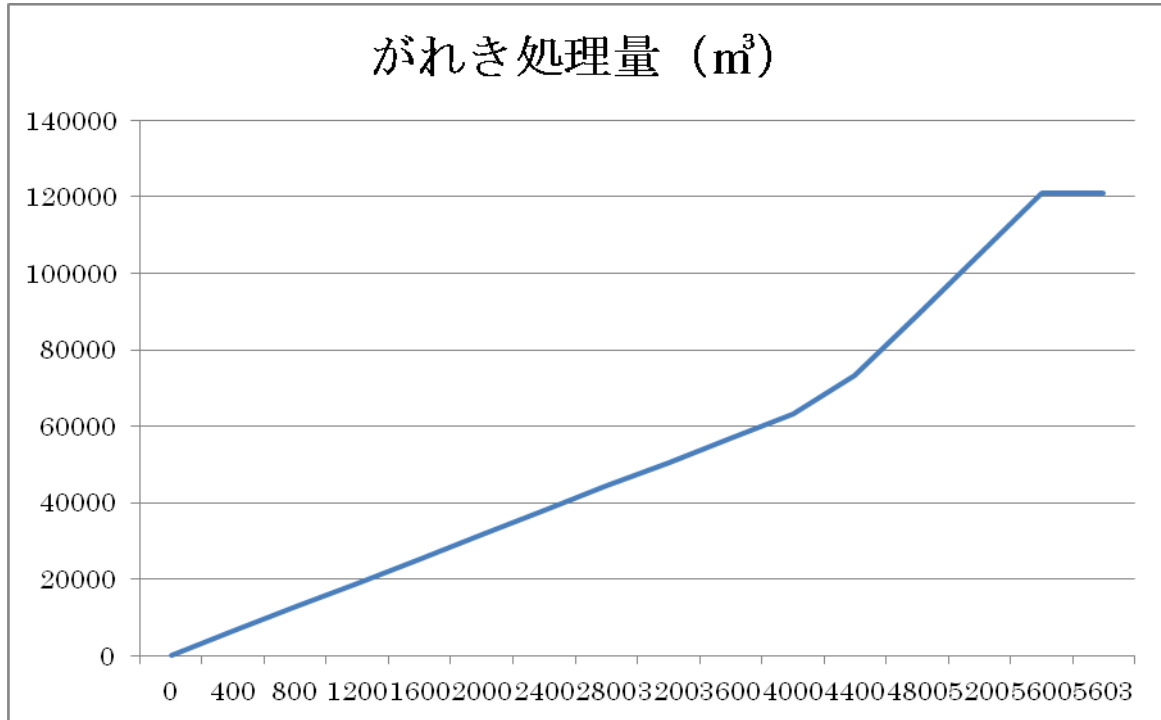
小山町



重機数は18台で一定。各モデル同じ挙動を示した。

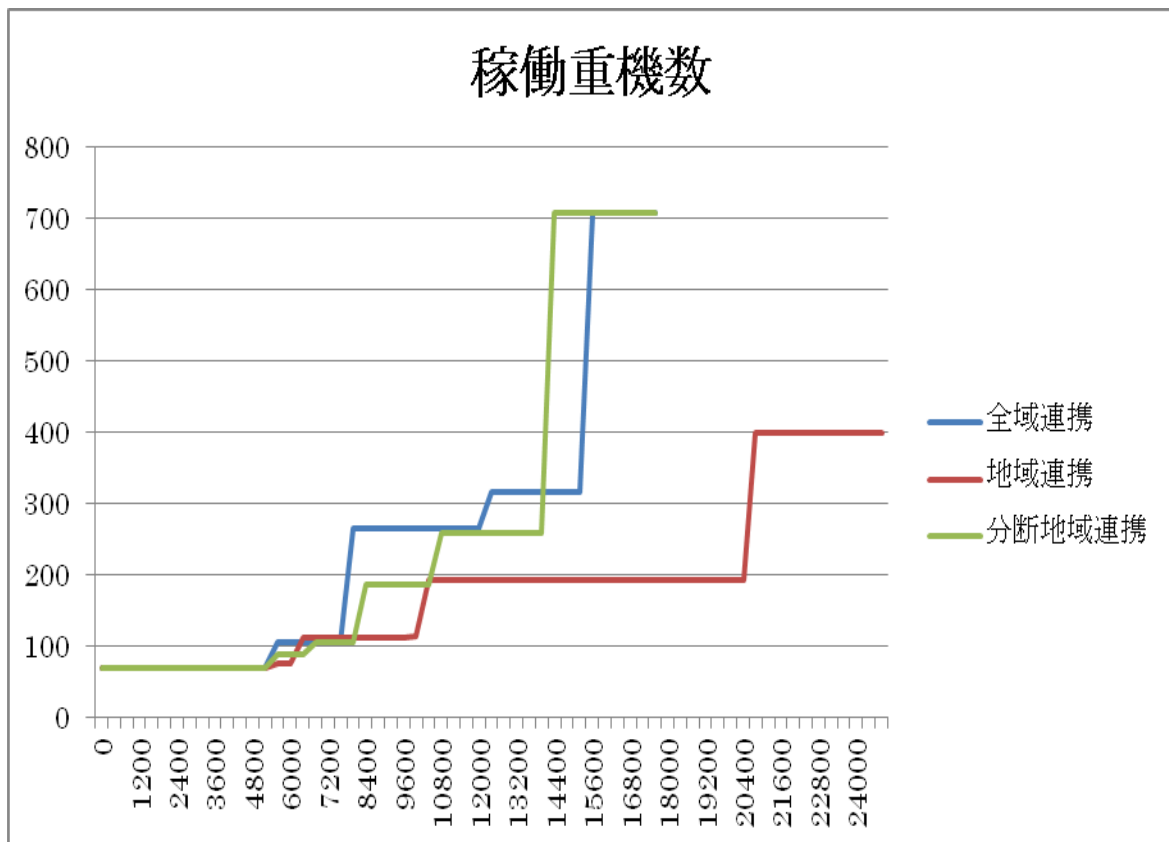
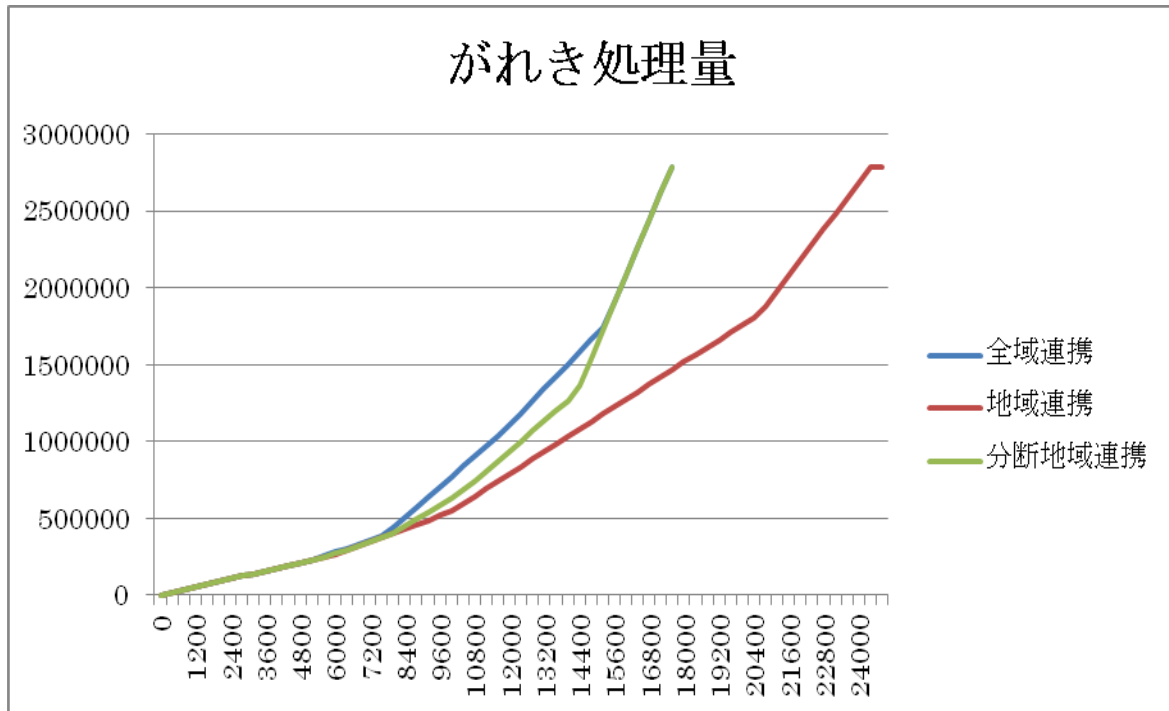


松崎町

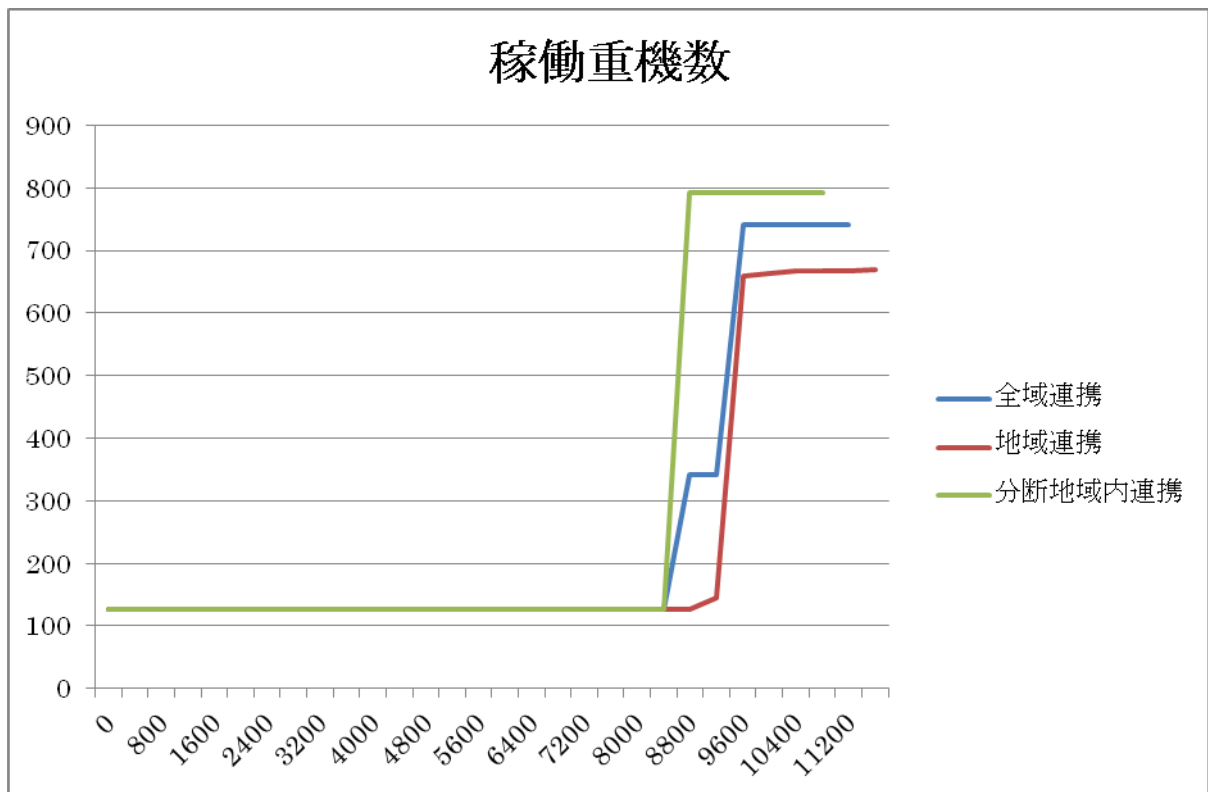
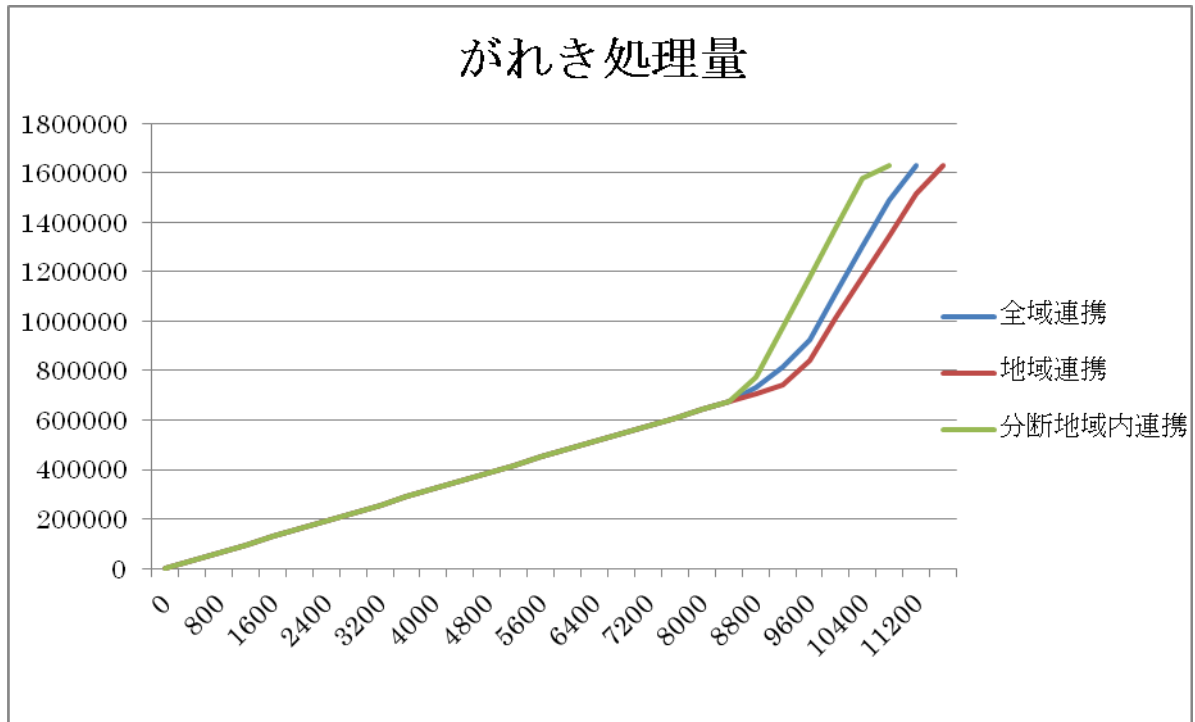


各モデル同じ挙動を示した.

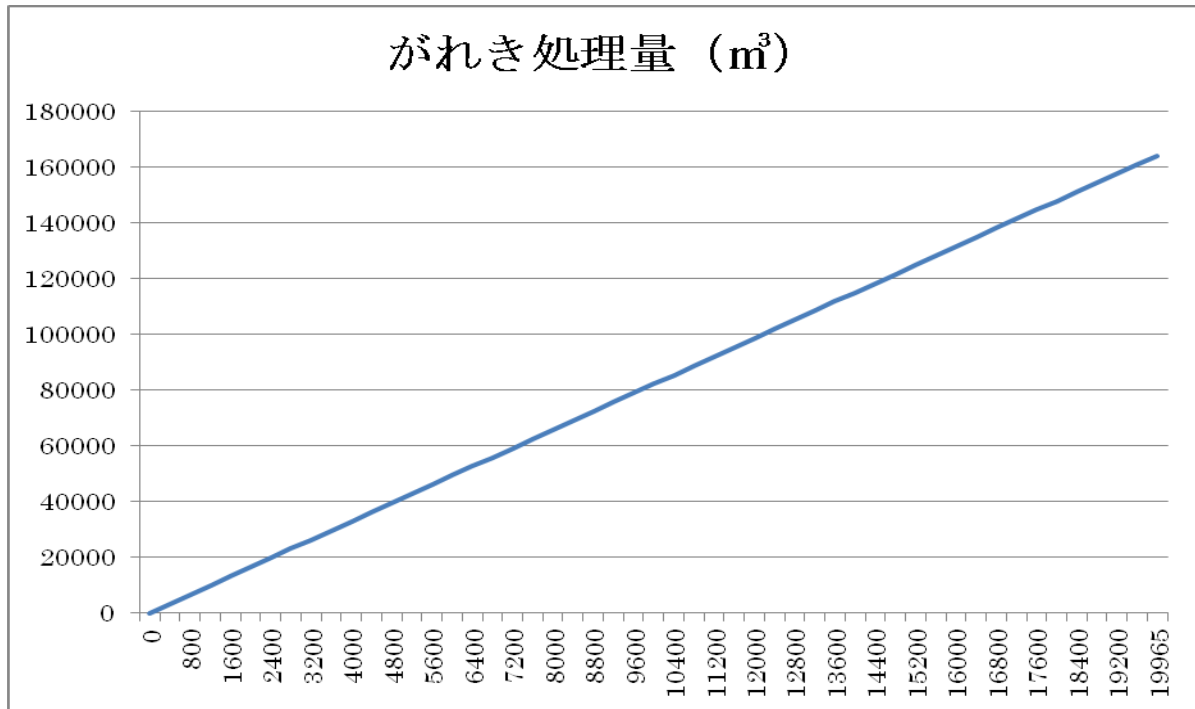
沼津市



焼津市

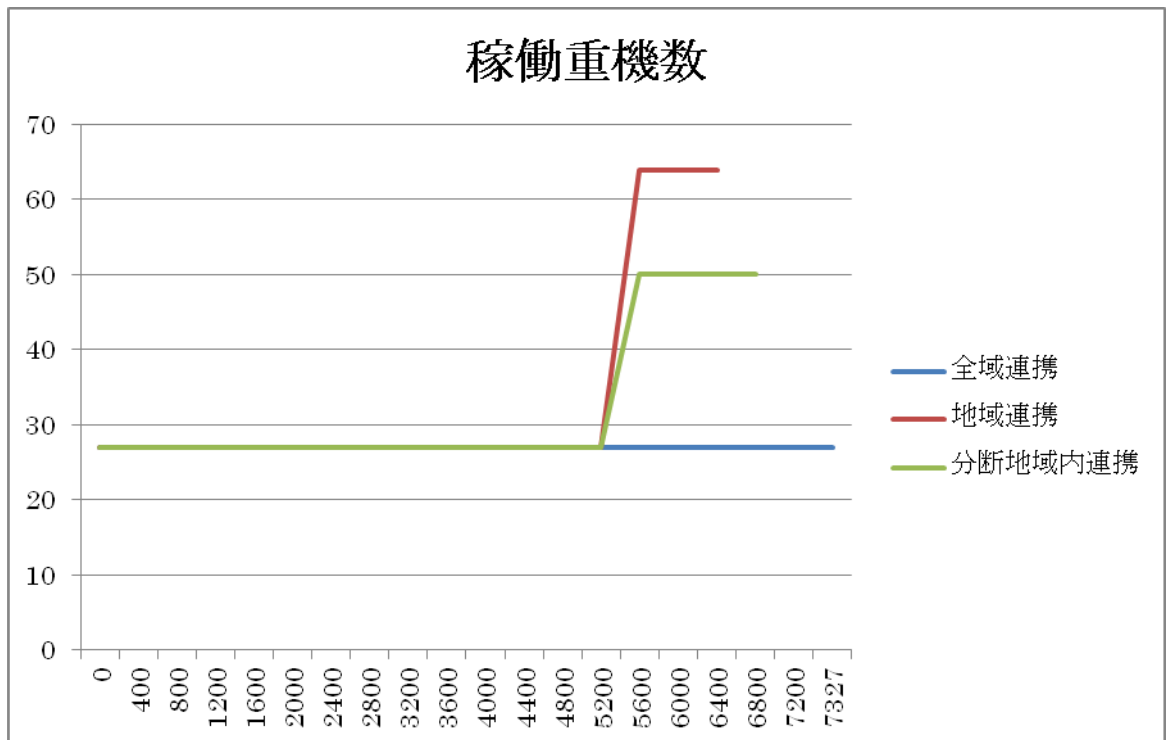
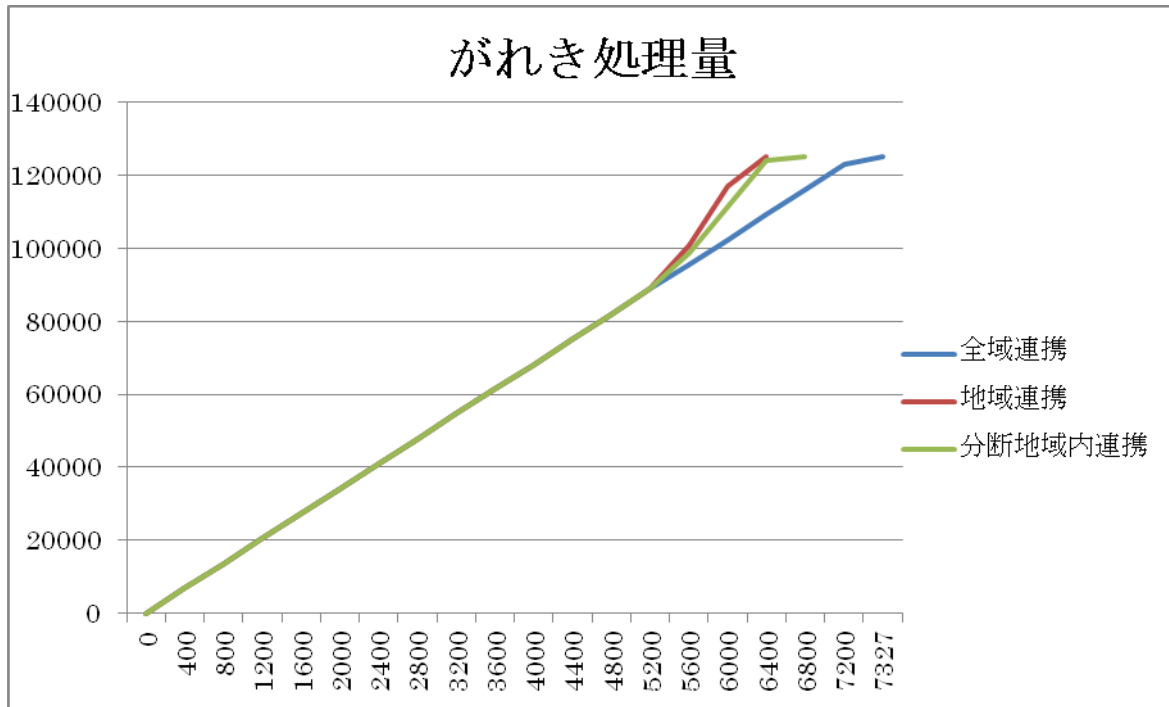


森町

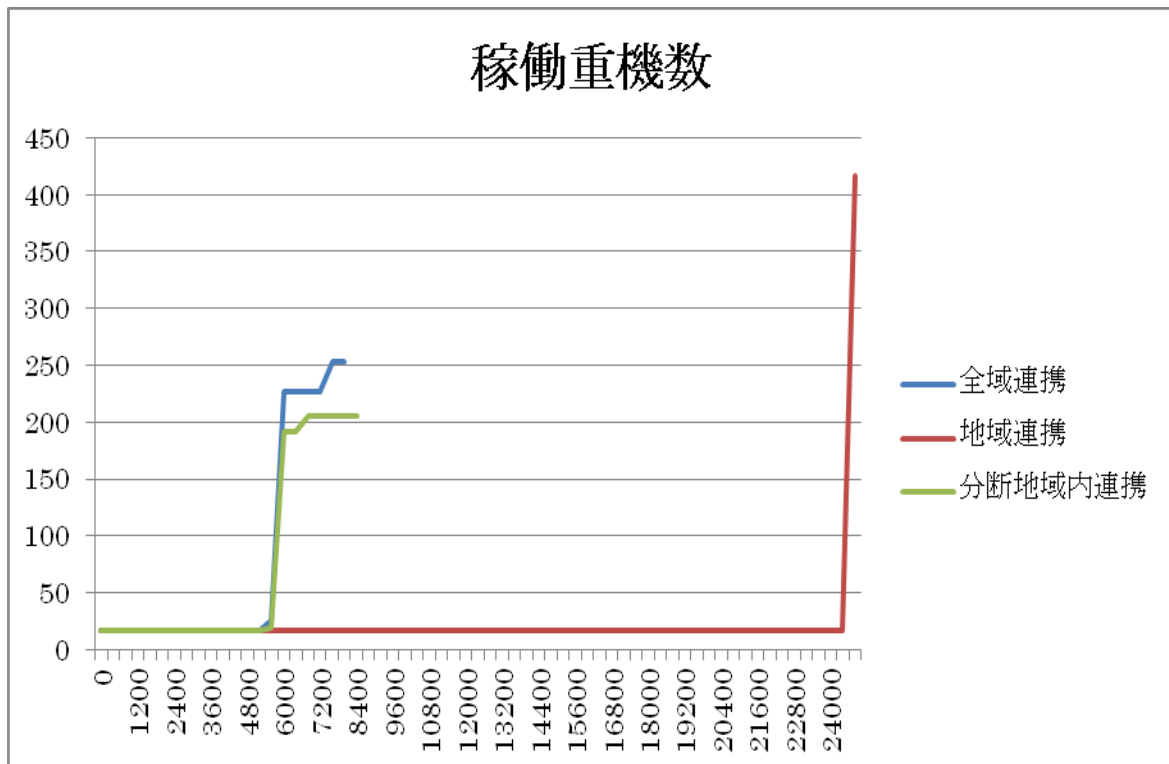
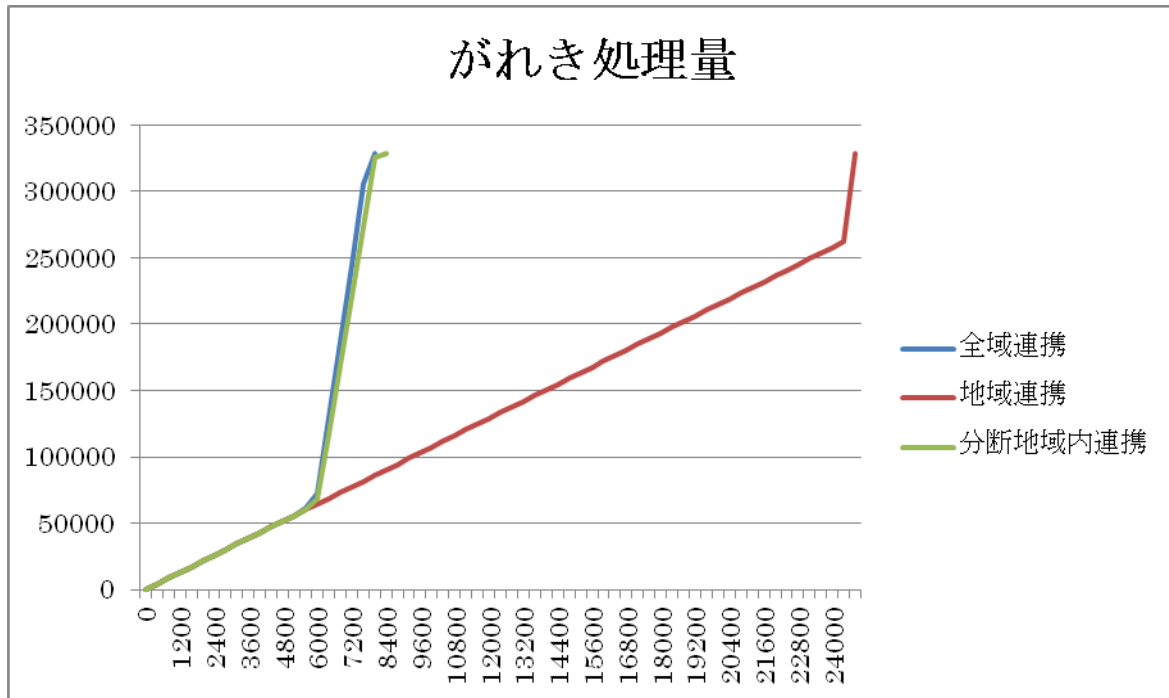


稼働重機数は 13 台で一定. 各モデル同じ挙動を示した.

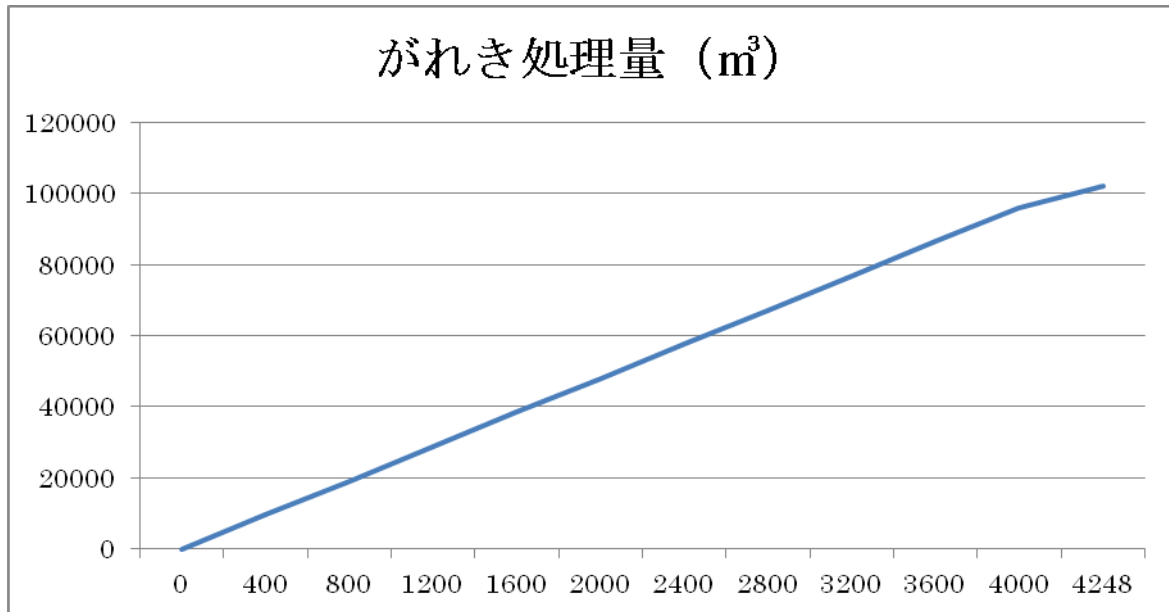
# 裾野市



清水町

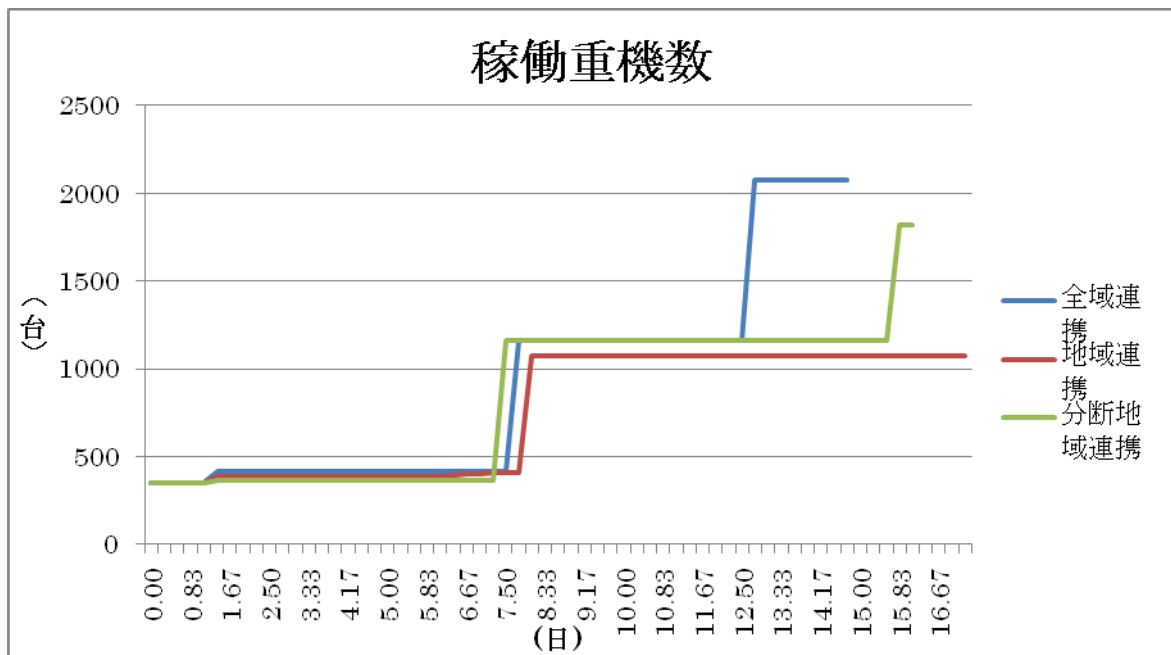
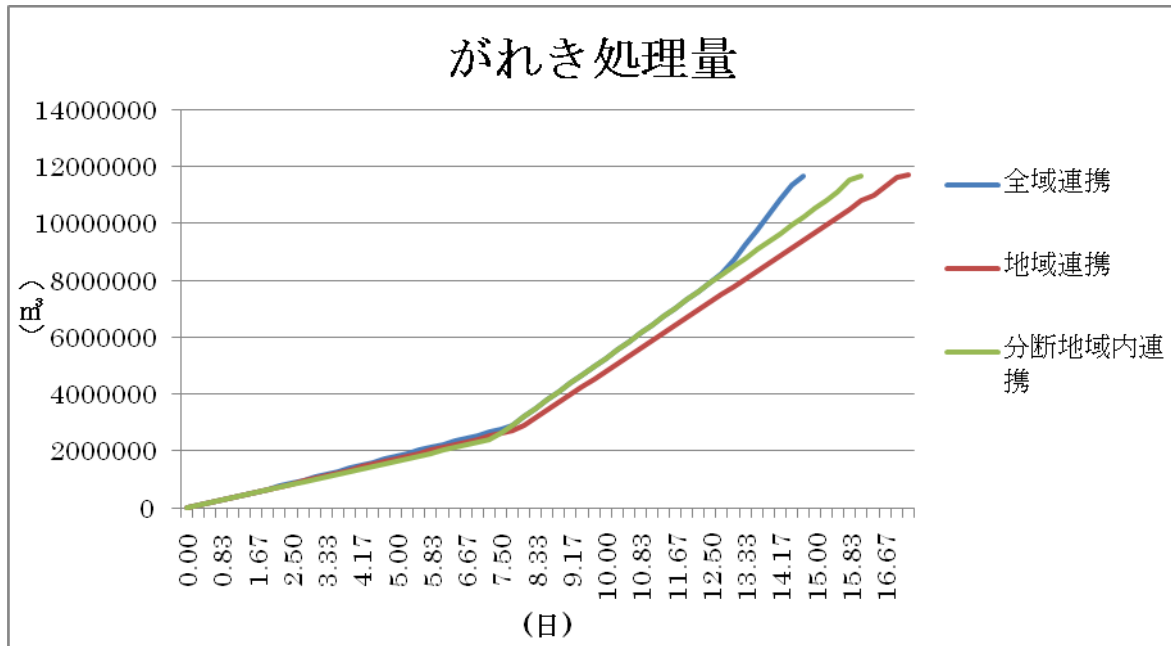


西伊豆町



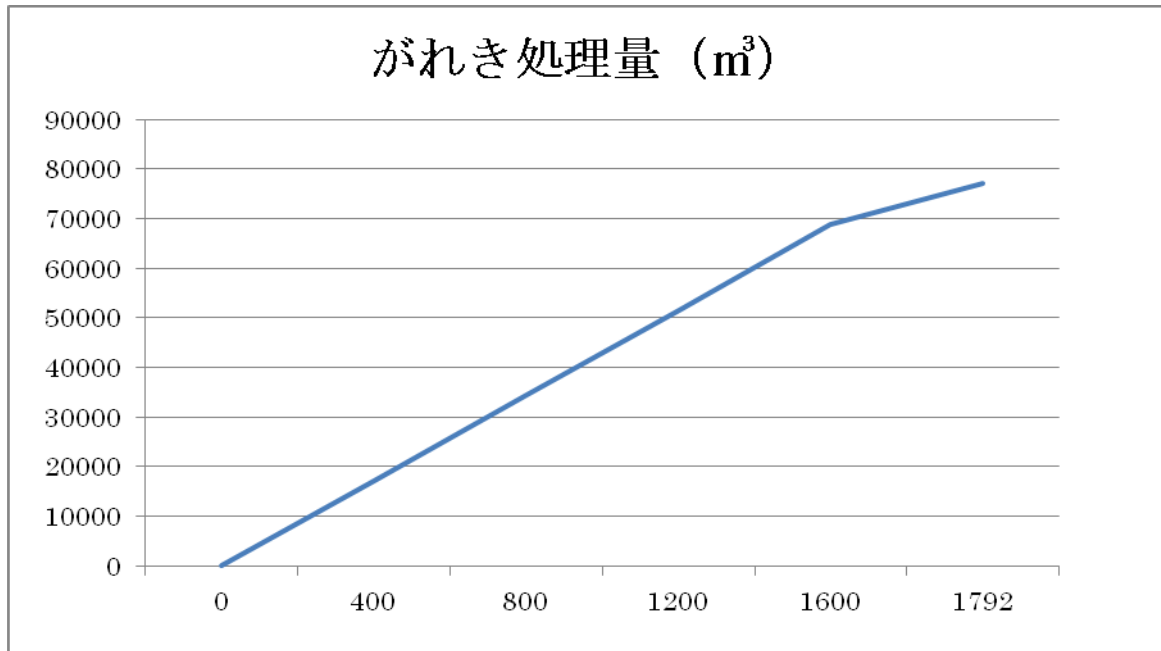
稼働重機数は 38 台で一定. 各モデル同じ挙動を示した.

静岡市



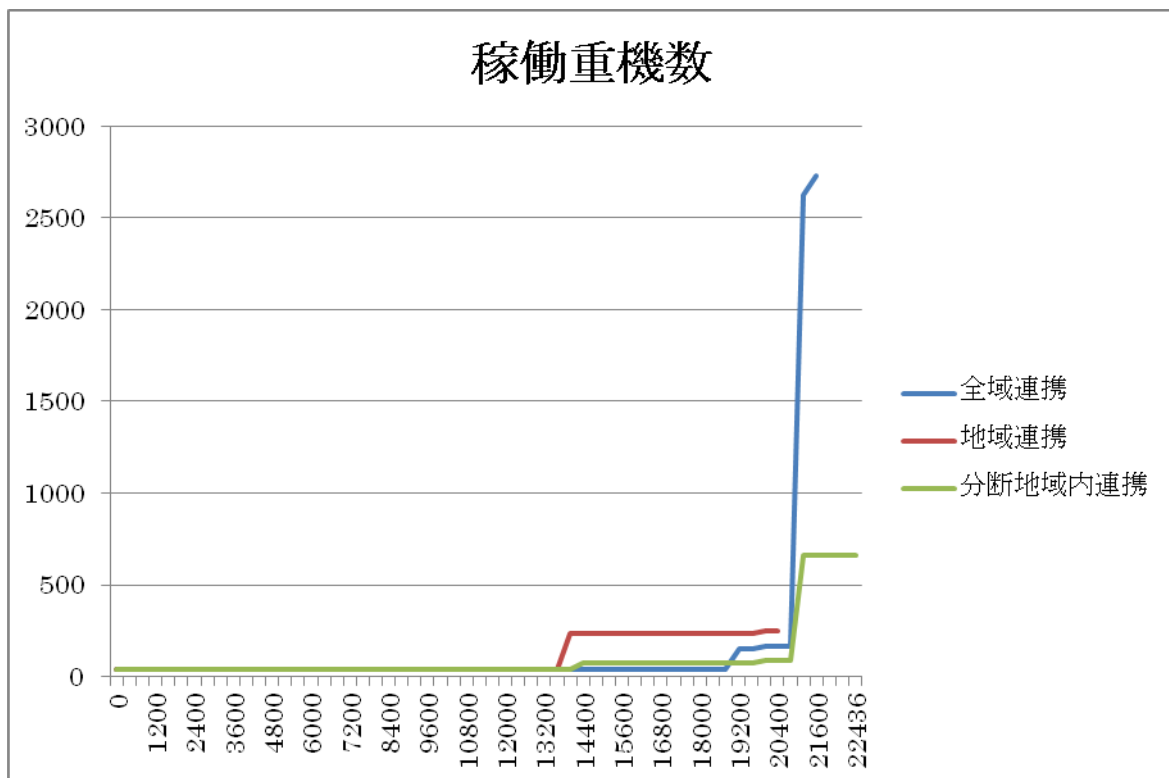
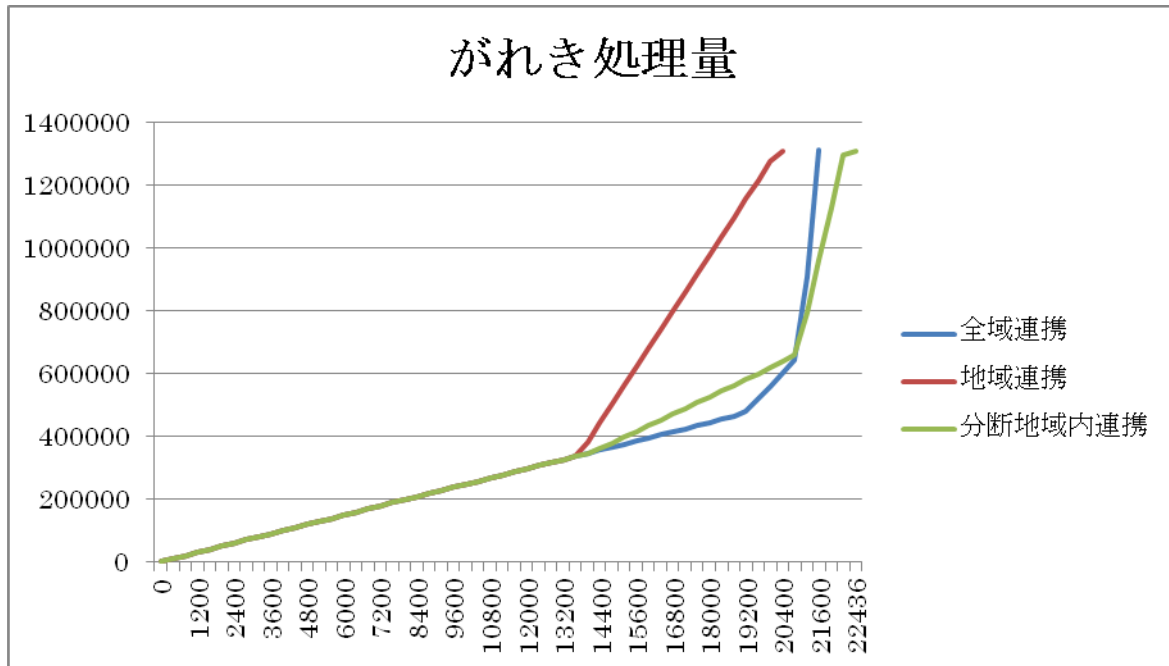


川根本町

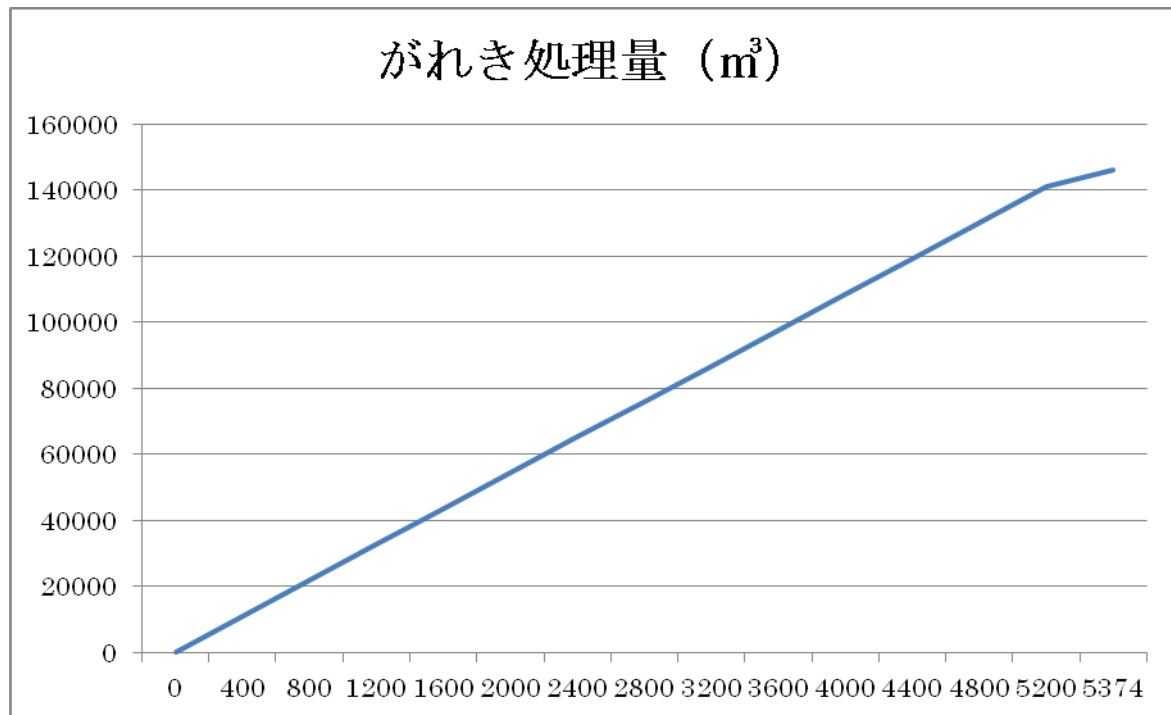


稼働重機数は 68 台で一定. 各モデル同じ挙動を示した.

袋井市

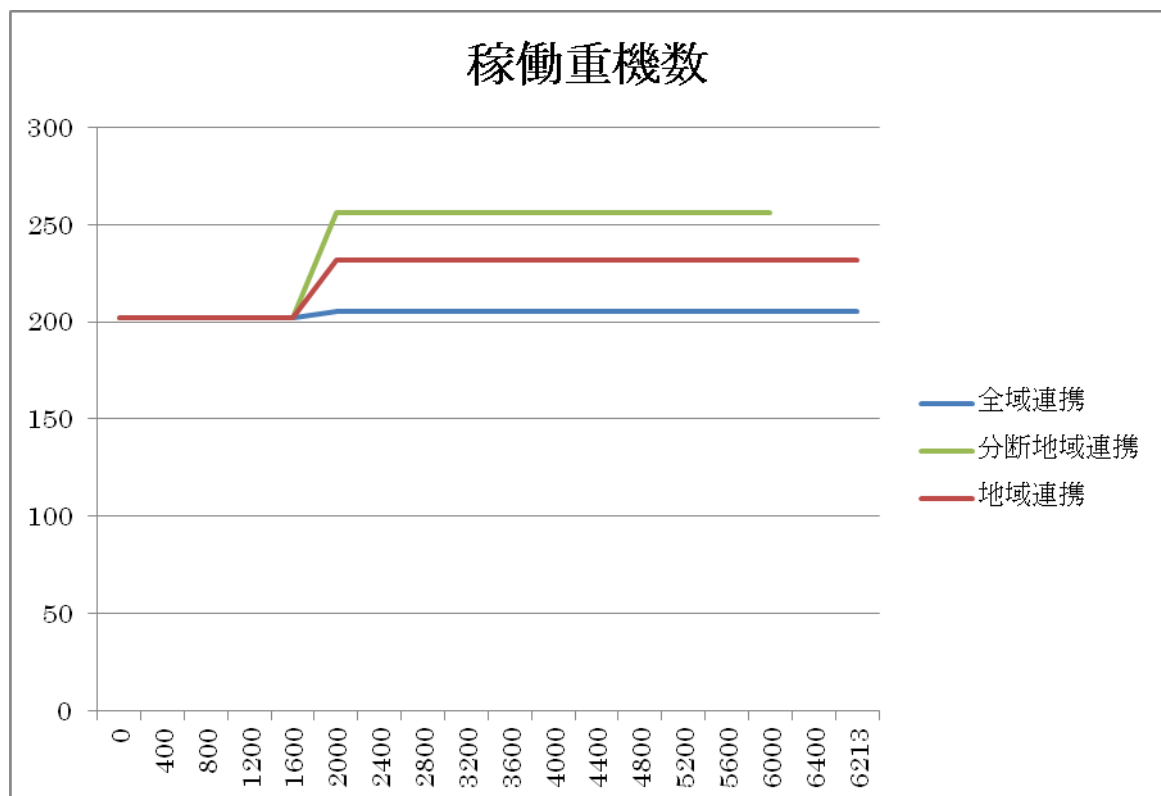
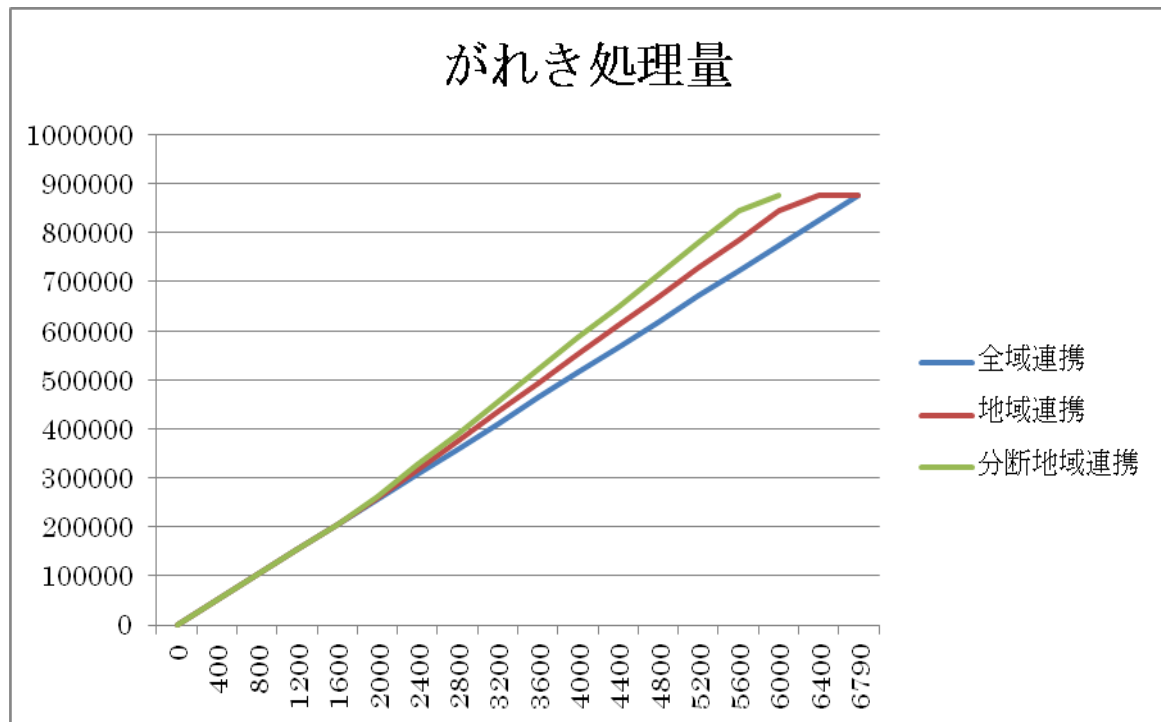


長泉町

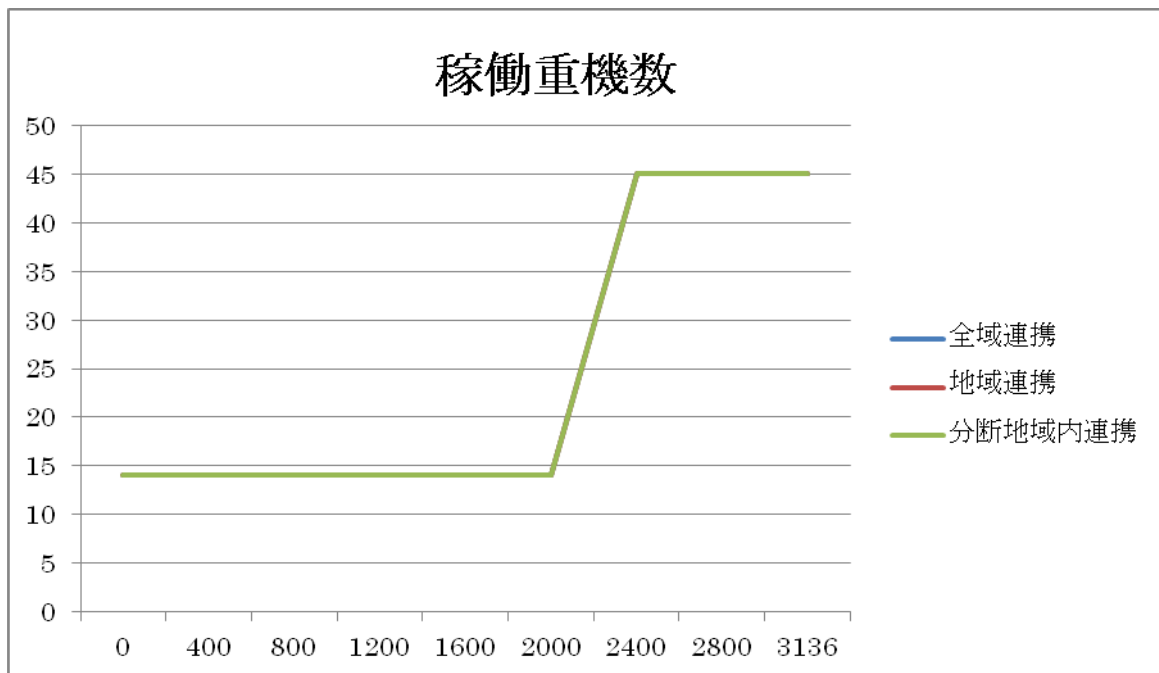
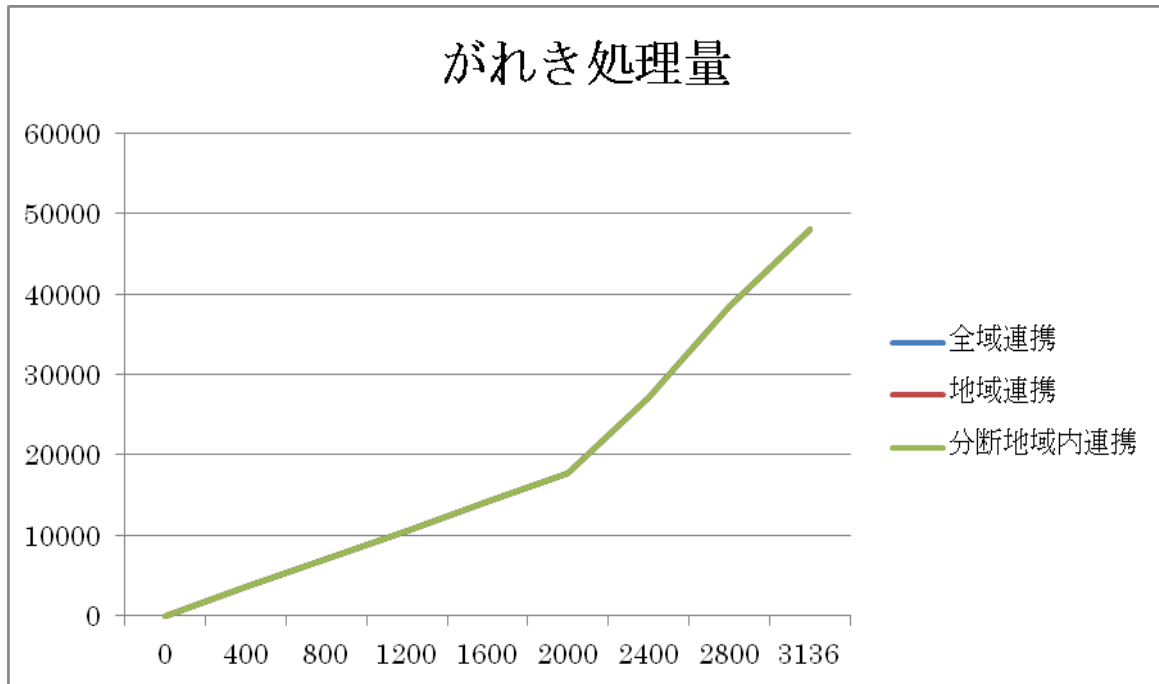


稼働重機数は 43 台で一定. 各モデル同じ挙動を示した.

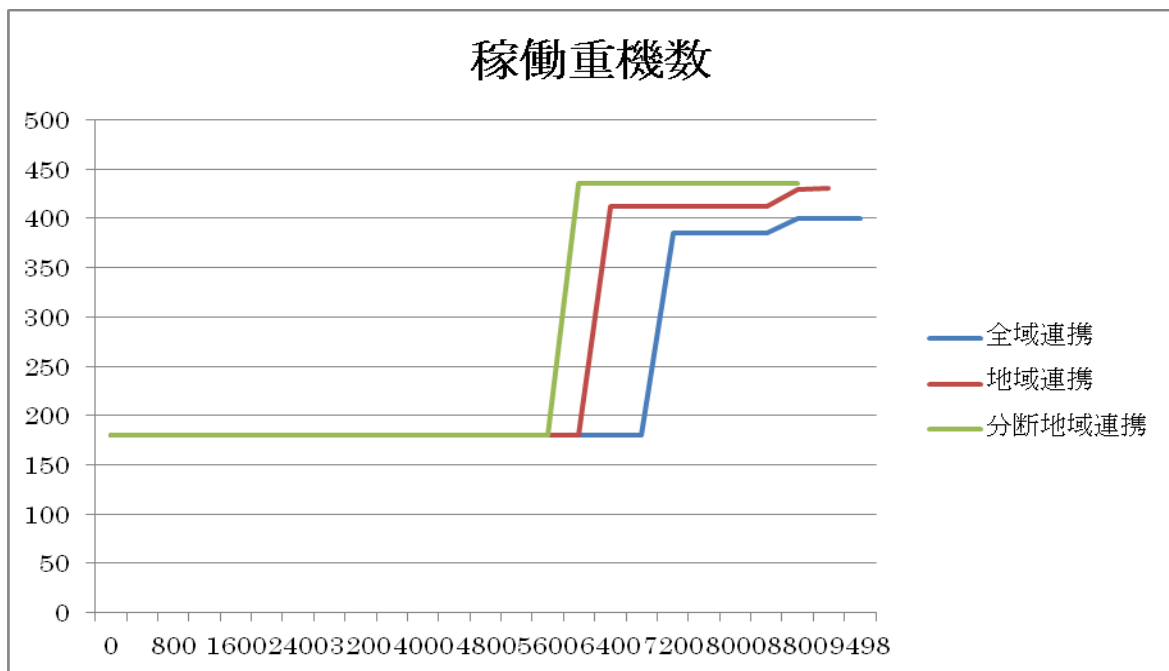
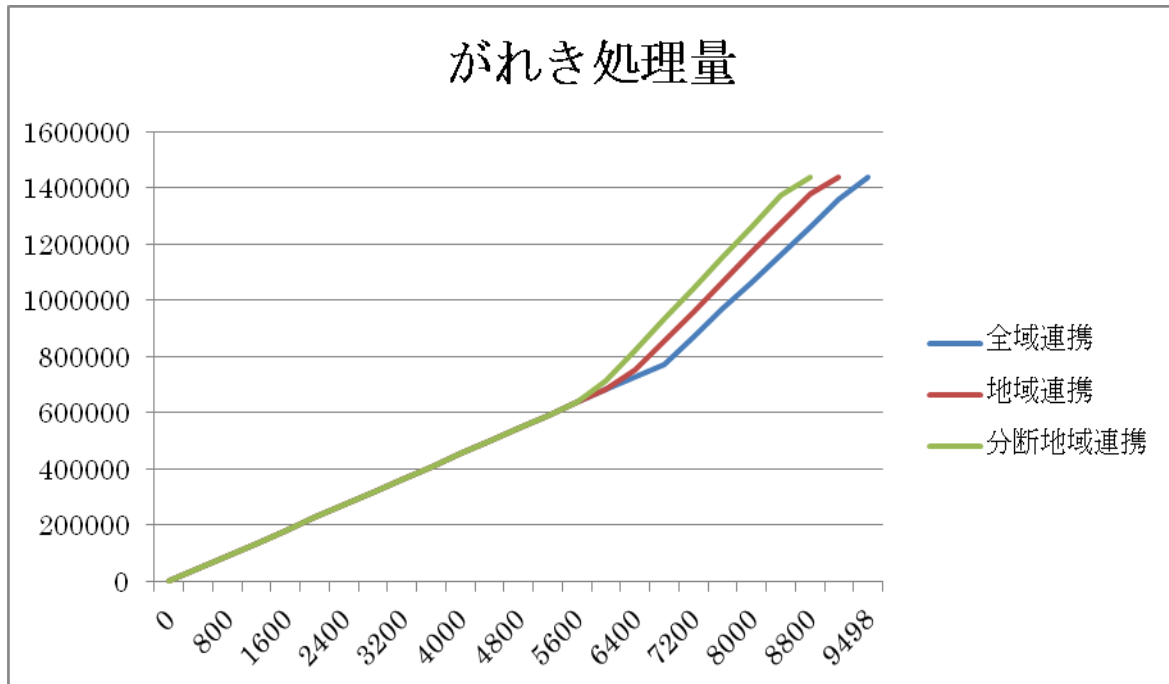
# 島田市



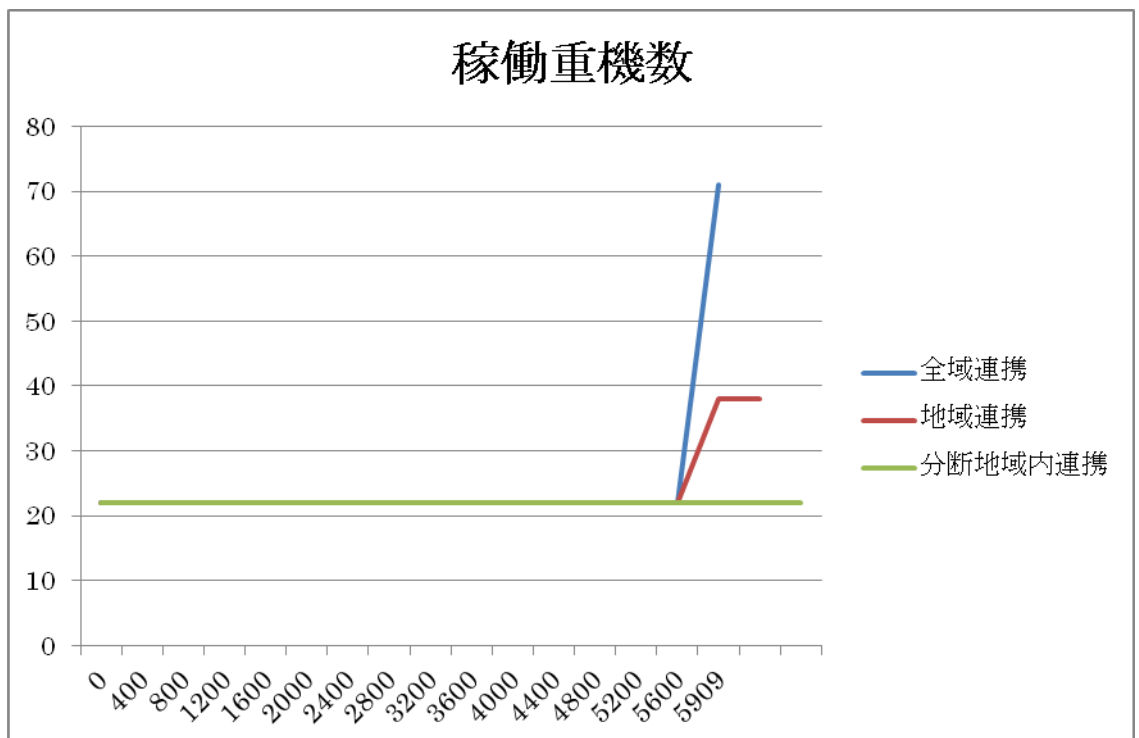
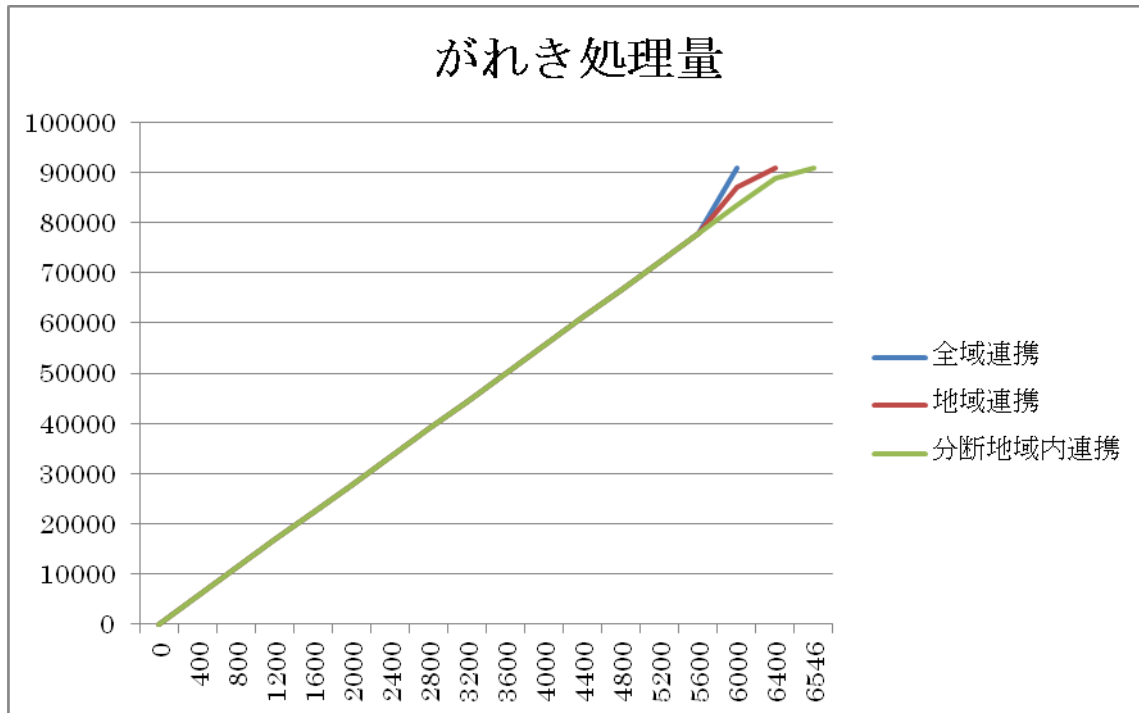
東伊豆町



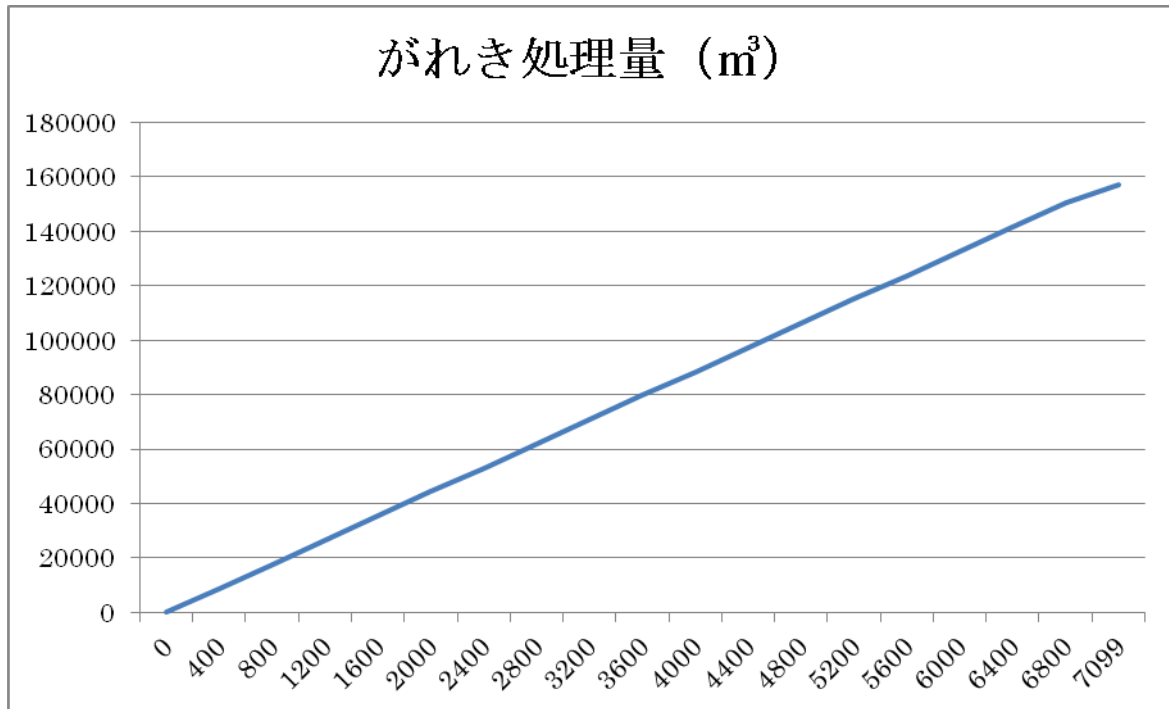
藤枝市



南伊豆町



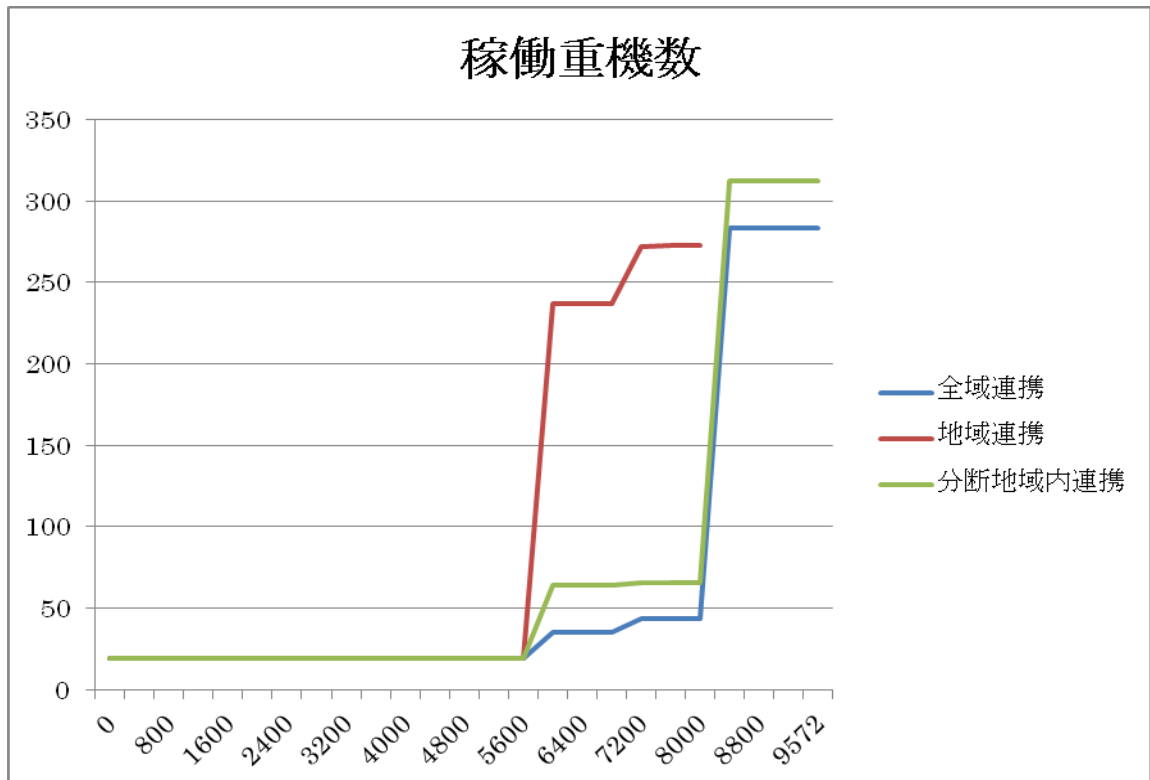
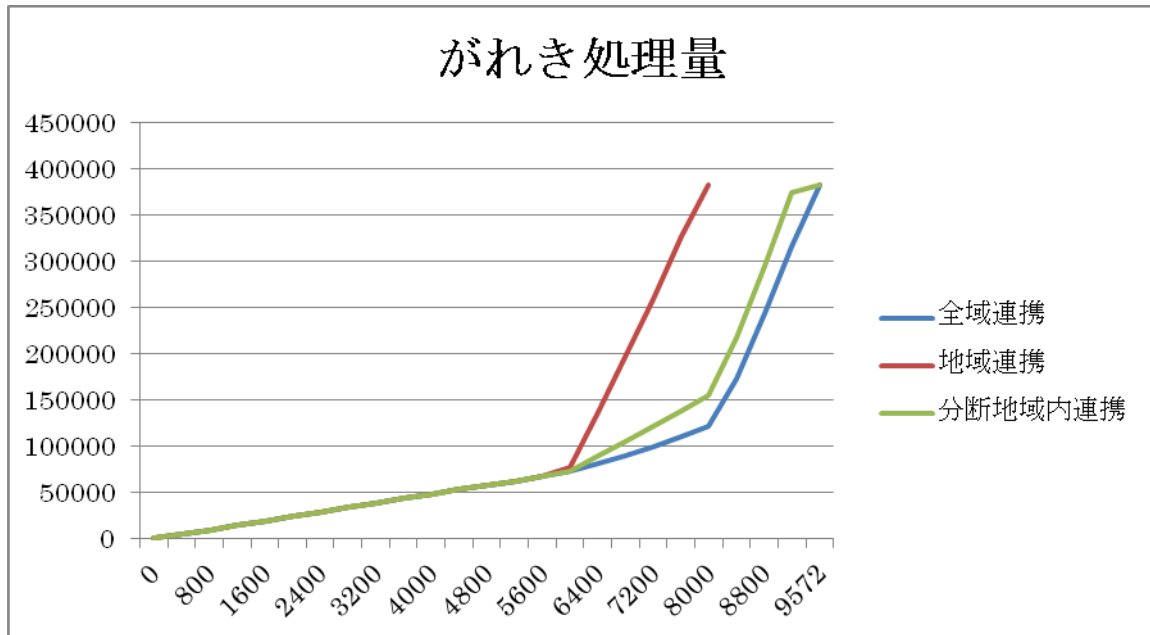
熱海市



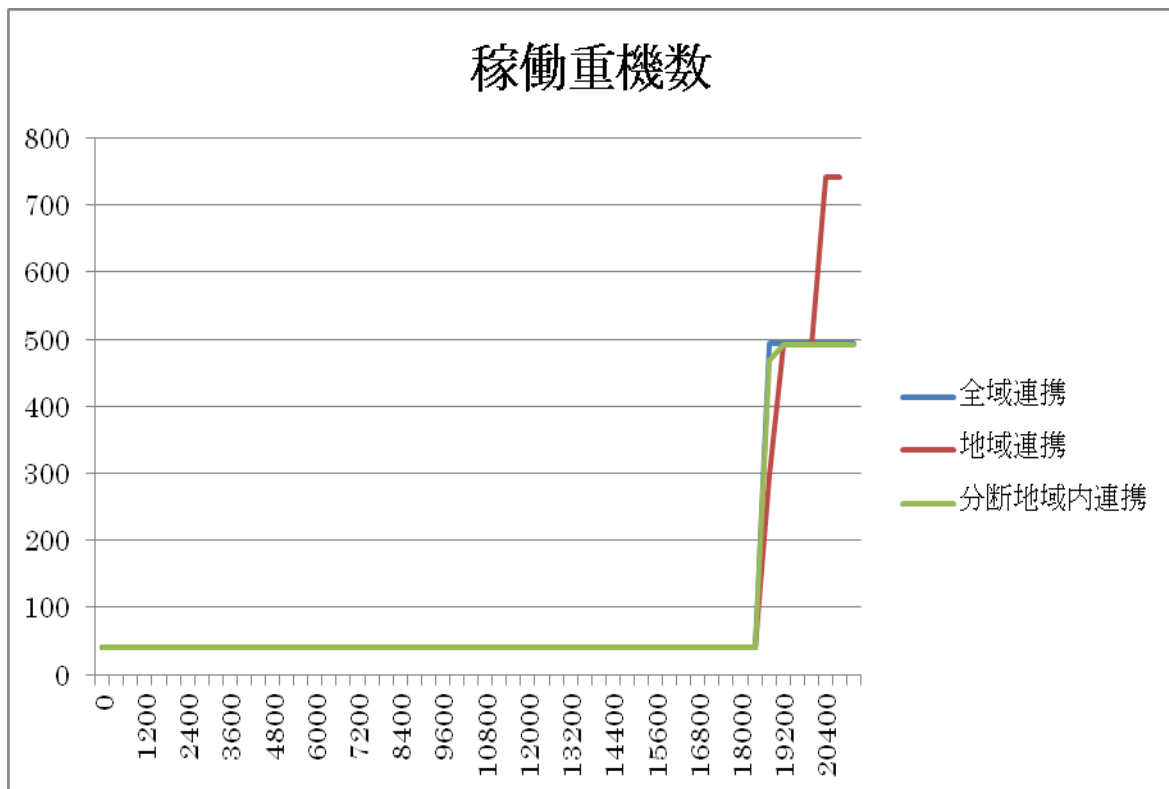
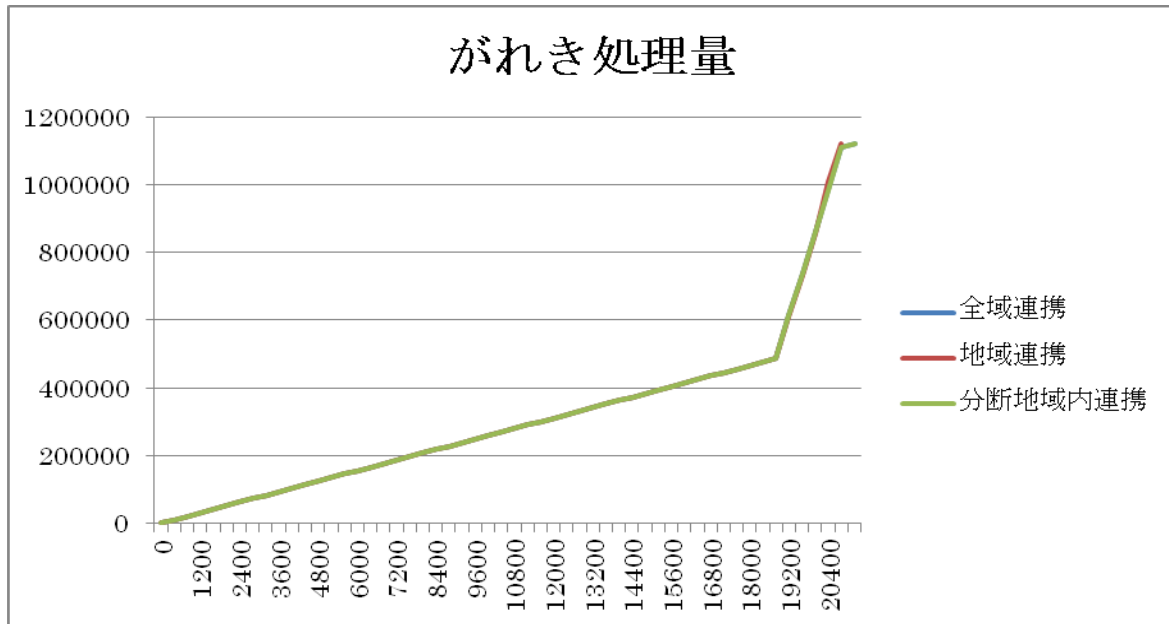
稼働重機数 35 台で一定. 全てのモデルで同じ挙動を示した.



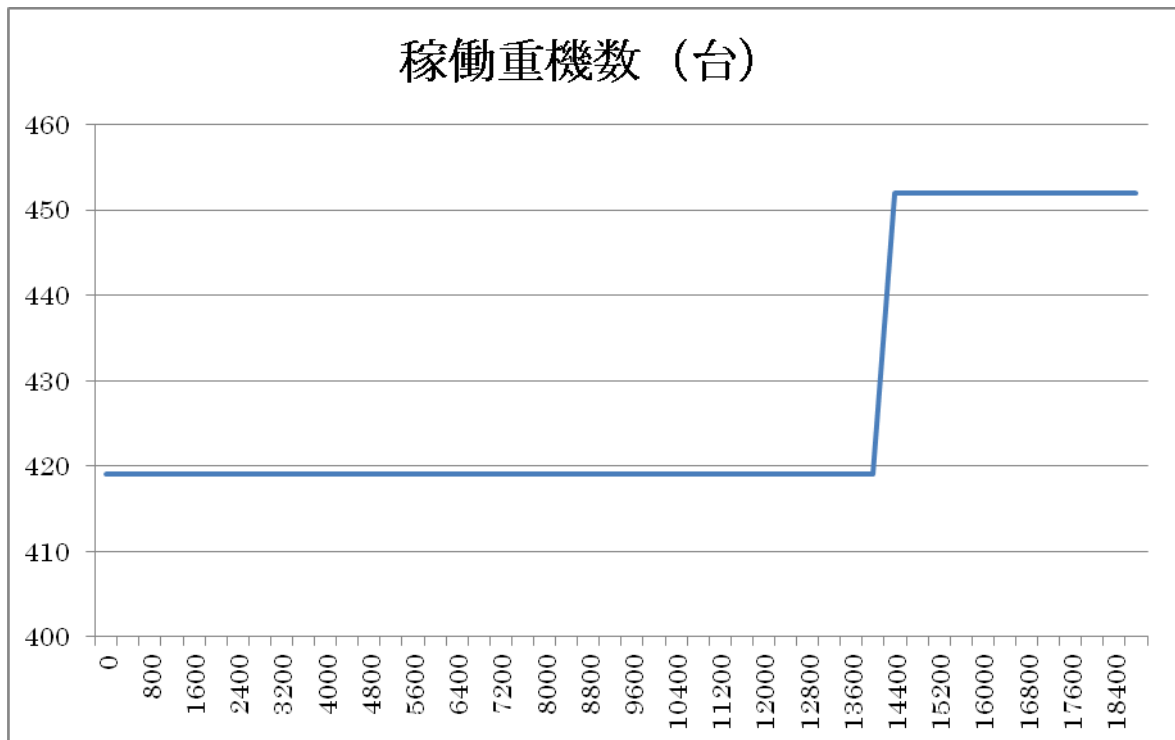
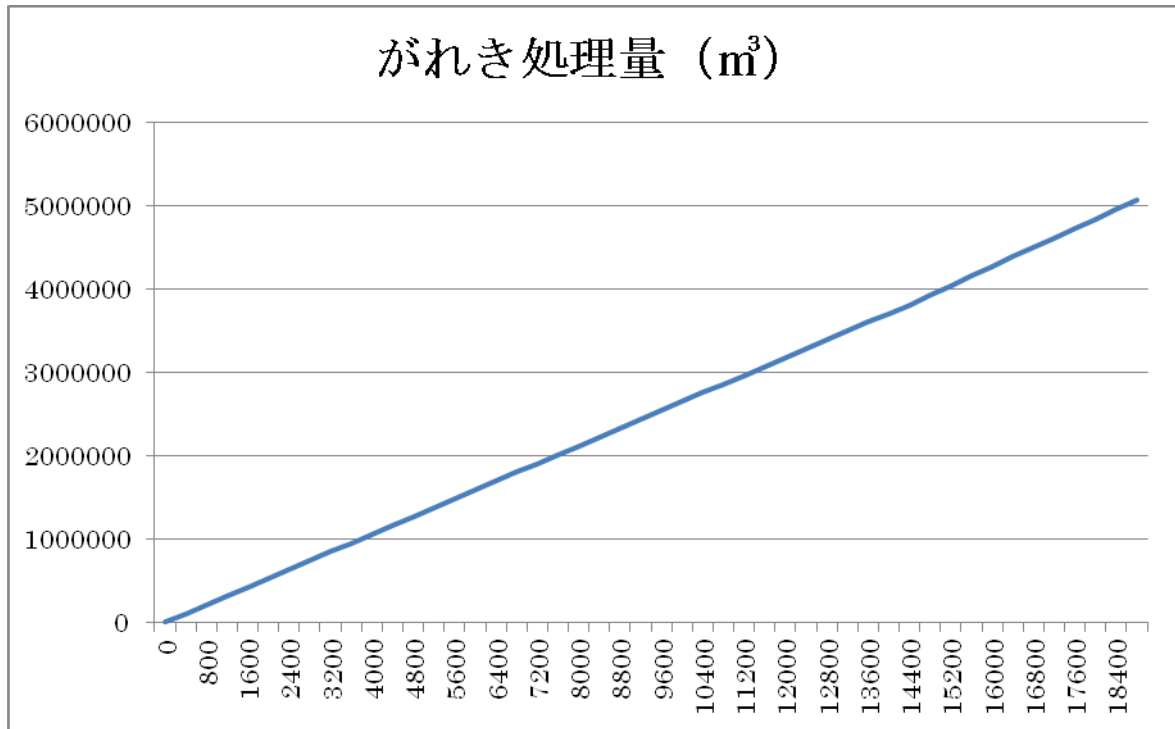
函南町



磐田市

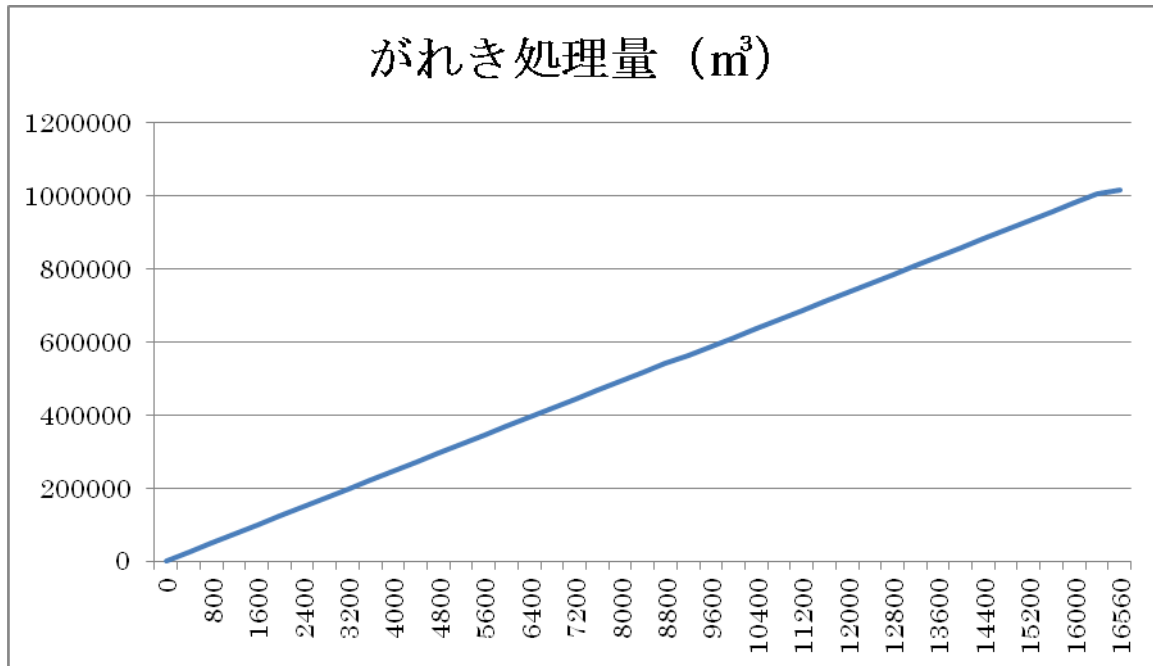


浜松市



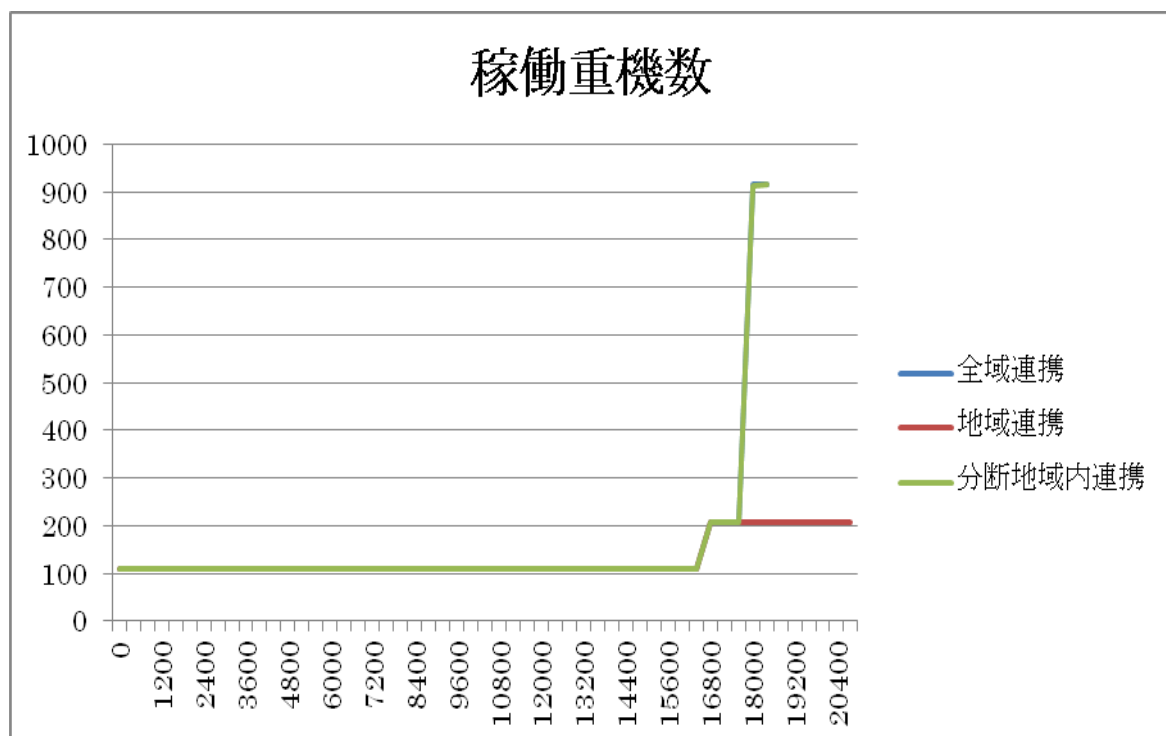
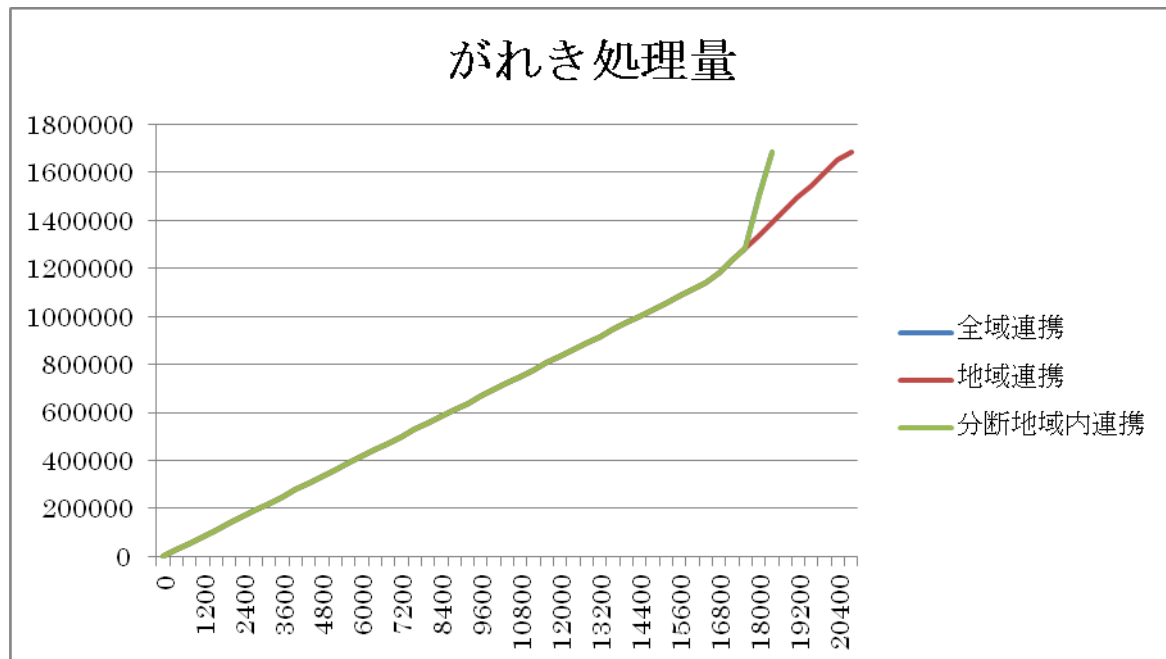
全てのモデルで同じ挙動を示した.

富士宮市

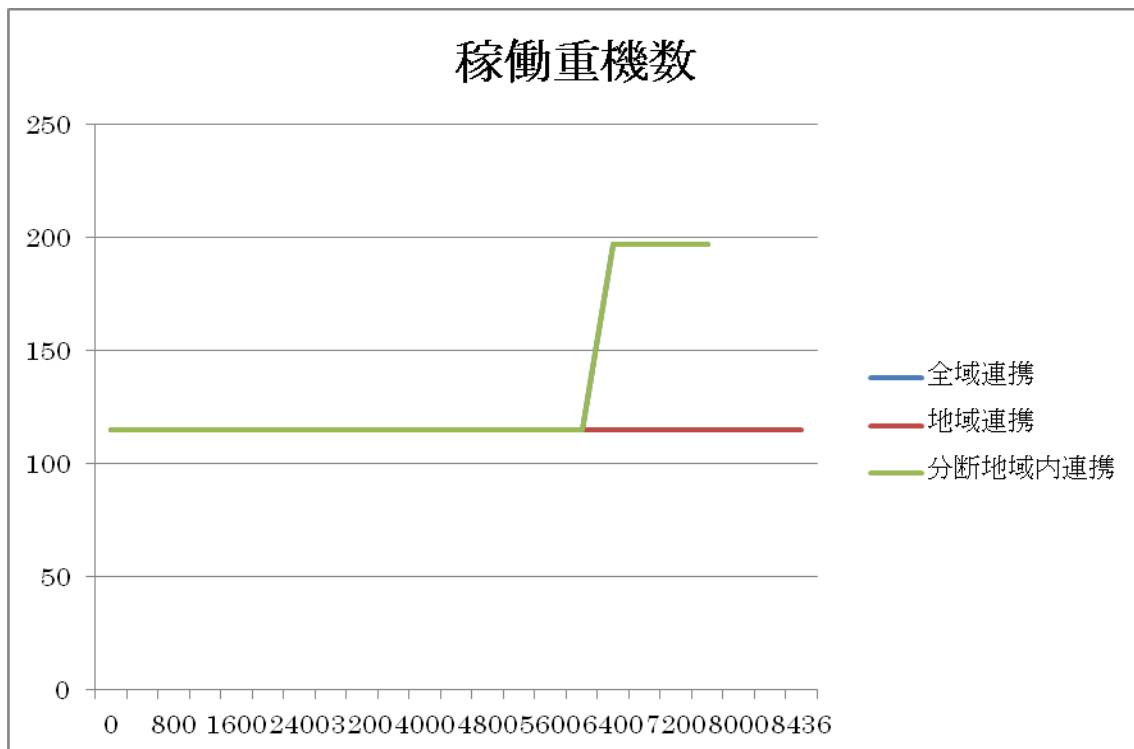
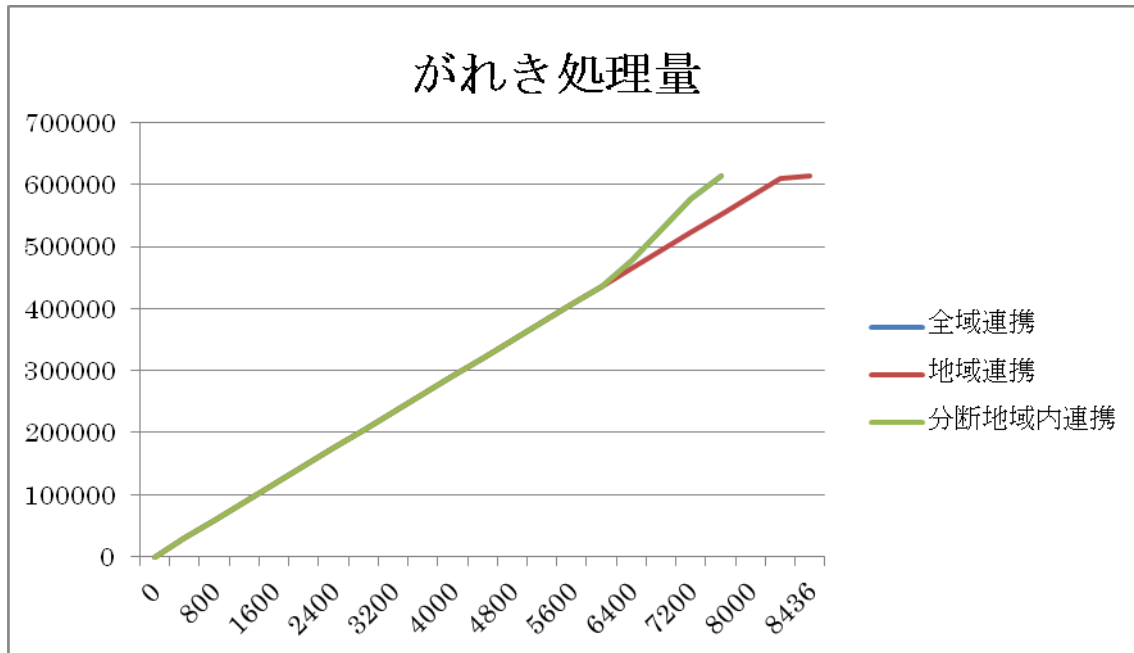


稼働重機数は 97 台で一定. 全てのモデルで同じ挙動を示した.

# 富士市



牧之原市



## 第 5 章

### 結論

## 5. 結論

地域防災力の低下が叫ばれる中、地元建設業者の保有する建設重機は大きな影響を与えている。しかし、重機の保有台数は減少する一方である。少ない重機数で効率的に作業には、地域間における重機の偏在の影響を少なくする事が必要であり、それには地域間における重機の共有体制が有効である。そこで、本研究では静岡県を対象に、広域での重機の共有の連携体制の有無が災害初期の啓開に対しどのような影響を及ぼすかをマルチエージェントシミュレーターを用いて検討を行った。

建設重機の調査を行った結果、重機の保有台数に関する詳細は存在せず、国や県レベルでの把握もできていない現状が分かった。しかし、静岡県庁交通基盤部の方々がこちらの問い合わせに対し、早急な対応をされ、調査、集計をして頂き、貴重な重機保有台数に関するデータを頂くまでに至った。そのデータをもとに広域での連携によるシミュレーションを行った。

シミュレーションの結果、地域間での連携が無い場合に比べ、連携がある場合では、がれきの処理日数が大幅に低下した。このことにより地域間連携の有効性を確認することができた。また、中規模地域での連携、緊急輸送路の被災状況を反映し、静岡県が東西に分断された場合の重機のがれき処理に与える影響も把握することができた。今回のシミュレーションでは重機の移動については渋滞を考慮せず、また多台数での重機の移動についても単独での移動と同じ時間とした。シミュレーション結果は県内の全がれきを利用可能な全重機で処理した場合の時間に重機の移動必要と考えられる時間を加えた値とほぼ合致したことから、本研究の設定する条件下において、その妥当性が確認された。今後、重機の移動に際しての渋滞、多台数での移動時における時間損失など、様々な条件下でのシミュレーションを行う場合に基礎となるモデルを確立できたと言える。

今後の課題としては、ダイクストラ法における局所最適解問題の解消、また、シミュレーションの効率化が求められる。



## 〔付録〕

### 1. 建設重機保有台数の推定について

本論では、静岡県内の建設業者が保有する重機のうち、市町別の災害協定に基づく災害派遣用の建設重機数のデータ静岡県交通基盤部の協力を得て取得した。これによると、静岡県全域の災害派遣用の建設重機数は2,733台であった。著者はこれに先立って、中部地方の建設重機数、中部地方の建設事業所数などの基礎的なデータを用いて、静岡県全体および県内市町の建設業者の保有重機数を推定した。

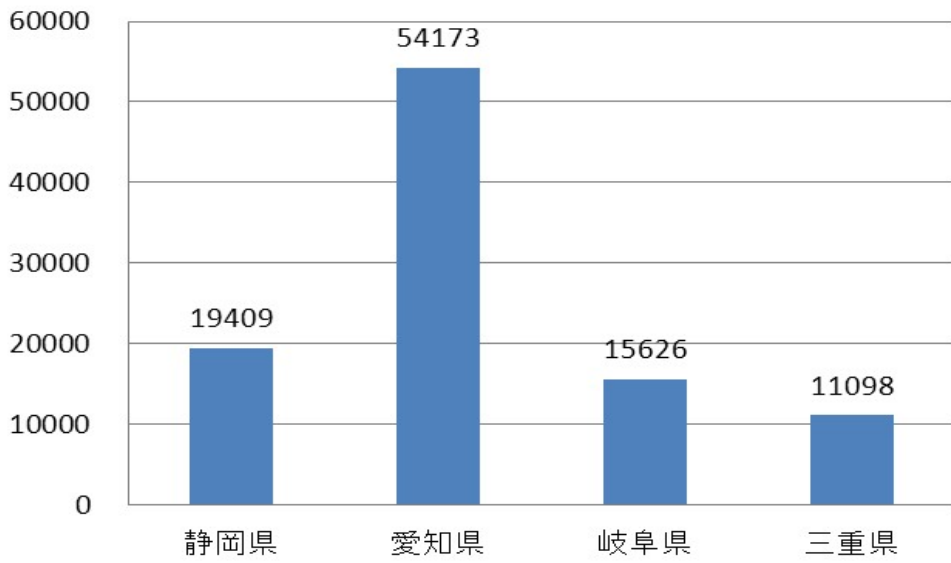
国土交通省では、隔年で建設重機動向調査を行っており、地方ごとの建設重機総数が公表されている。これによれば、中部地方の建設重機数は94,472台である。

付表-1は、静岡県が公表している経済センサスから得られた事業所数などの基礎データである。国土交通省からは県ごとの重機数が公表されていないことから、これらの基礎データから中部地方内の県ごとの建設重機台数を推定した。その結果を付図-1に示す。いずれのデータを用いるかによって、当然ながら推定される重機数は異なるが、ここでは、4ケースを平均している。

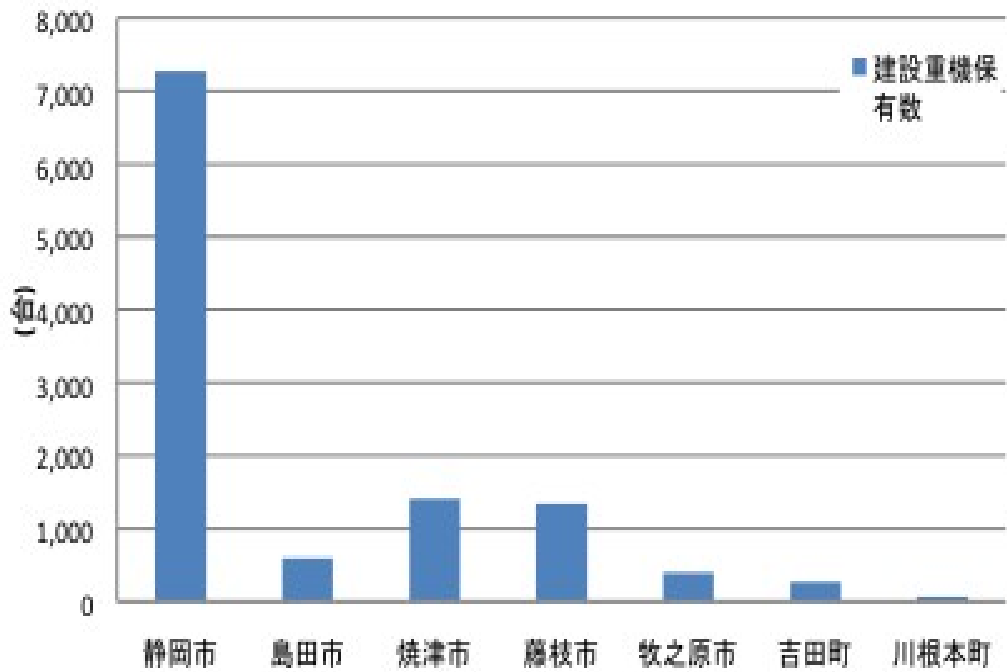
次に、この静岡県全体の建設重機数19,409台を用いて静岡県内市町の建設重機数を推定した。推定に当たっては、各市町内の建設事業所数に建設重機数が比例するものと仮定した。その結果を付図-2～付図-5に県内の4地域別に示す。この結果、浜松市や静岡市では、災害派遣用の建設重機数の10倍～20倍の重機が保有されていることになる。本研究では幸いに、県の全面的な協力を得て災害派遣重機数を把握することができたが、今後、全国的に災害時に利用可能な建設重機の把握をすることが必要であると考えらる。

付表-1 経済センサスから得られた基礎データ

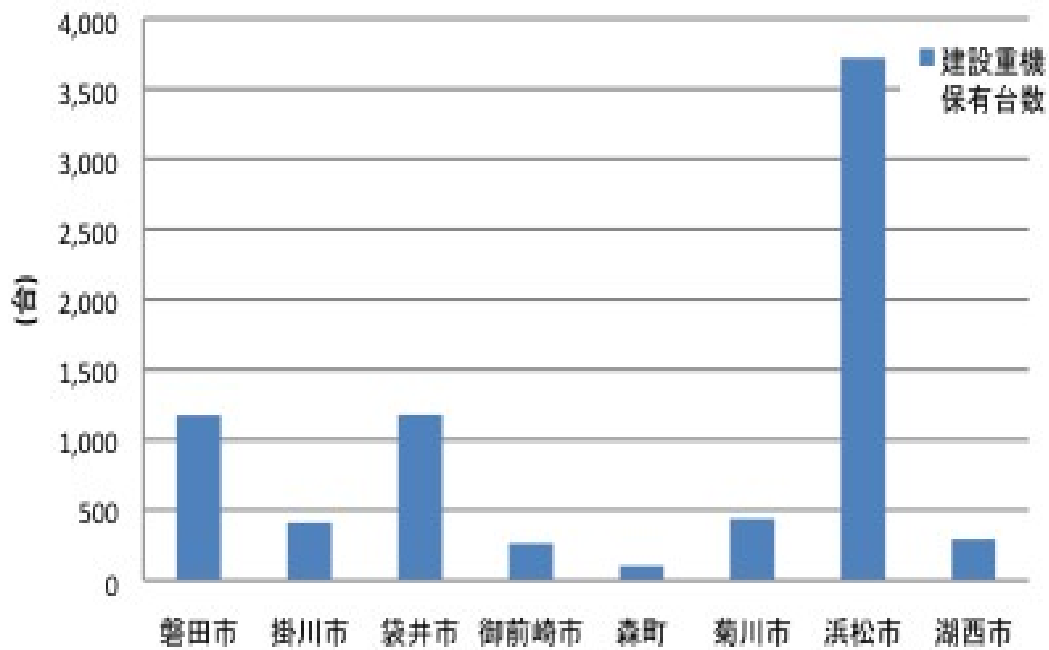
	事業所数	比率	従業者数	比率	土木工事元請け受注高(百万円)	比率	土木工事下請け受注高(百万円)	比率
静岡県	20,130	0.28	121,581	0.24	142,899	0.17	50,544	0.12
愛知県	28,691	0.40	243,016	0.48	486,855	0.56	266,522	0.66
岐阜県	12,353	0.17	77,069	0.15	140,944	0.16	54,613	0.13
三重県	9,714	0.14	62,502	0.12	92,484	0.11	34,535	0.09



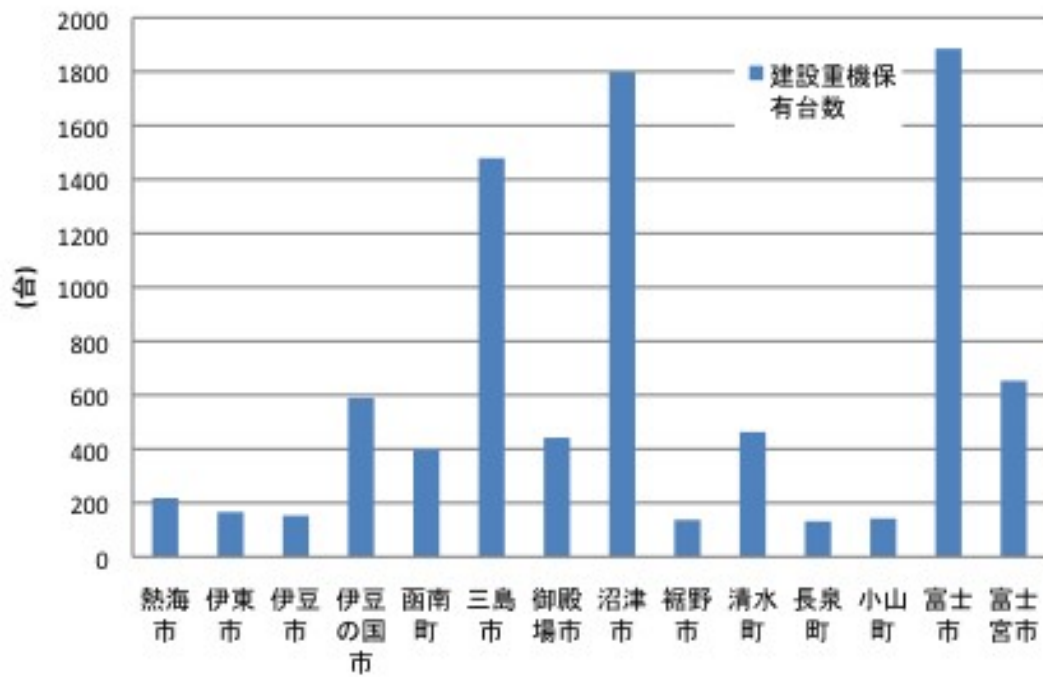
付図-1 中部地方の保有重機数の推定



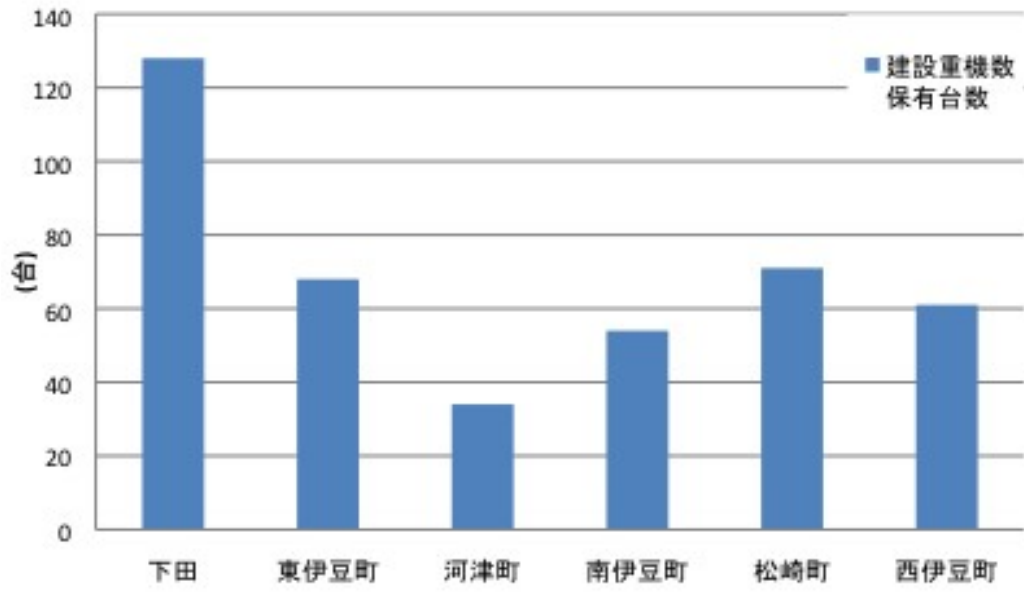
付図-2 中部地域の保有重機の推定



付図-3 西部地域の保有重機の推定



付図-4 東部地域の保有重機の推定



付図-5 賀茂地域の保有重機の推定

ロボット（バックホウ）エージェントルール

Agt\_Init{

my.Direction = Rnd() \* 360

my.speed = 0

dim i as double

i = rnd()

if i < universe.優先派遣割合 then

my.busy = true

else

my.busy = false

end if

my.shiya = 5

my.targetID = 9999

my.targetID2 = 9999

my.targetX = 9999

my.targetX2 = 9999

my.targetY = 9999

my.targetY2 = 9999

my.count = 0

}

Agt\_Step{

Dim one As Agt

dim one2 as agt

Dim markbox As Agtset

Dim markbox2 as Agtset

dim yusen as agtset

Dim exitbox As Agtset

Dim exitbox2 As Agtset

Dim targetID As Integer

Dim targetID2 As Integer

Dim targetX As Integer

Dim targetX2 As Integer

Dim targetY As Integer

Dim targetY2 As Integer

dim tomaru as agtset

dim tomaru2 as agtset

dim robox as agtset

dim robox2 as agtset

makeoneagtsetarounddown(tomaru,2,universe.二次元空間.本部,false)

makeoneagtsetarounddown(tomaru2,2,universe.二次元空間.優先本部,false)

if GetCountStep() >15 then

my.speed = 1

if my.busy == false then

if countagtset(tomaru) == 1 then

my.speed = 0

else my.speed = 1

end if

else

if countagtset(tomaru2) == 1 then

my.speed = 0

else my.speed = 1

```
end if
end if
end if
```

```
MakeOneAgtsetAroundOwn(markbox, my.shiya, Universe.二次元空間.マーカー, False) //視野範囲にいるマーカーを markbox に格納する
```

```
MakeOneAgtsetAroundOwn(exitbox, 7, Universe.二次元空間.本部, False) //視野範囲にいる本部を exitbox に格納する
```

```
MakeOneAgtsetAroundOwn(markbox2, my.shiya, Universe.二次元空間.マーカー, False) //視野範囲にいるマーカーを markbox2 に格納する
```

```
MakeOneAgtsetAroundOwn(exitbox2, 7, Universe.二次元空間.優先本部, False) //視野範囲にいる優先本部を exitbox2 に格納する
```

```
MakeOneAgtsetAroundOwn(yusen, 1, Universe.二次元空間.マーカー, False)
```

```
if markbox == 0 then
killagt(my)
end if
```

```
if my.busy == false then
my.color = color_red
else
my.color = color_green
end if
```

```
if my.busy == False then      //通常モード
for each one in markbox2
my.targetID = one.MinID      //目的マーカーの ID,X,Y を保存
my.targetX = one.MinX
my.targetY = one.MinY
my.shiya = 3
next one
end if
```

```
if my.busy == true then      //優先モード
for each one2 in markbox2
my.targetID2 = one2.MinID2   //目的マーカーの ID,X,Y を保存
my.targetX2 = one2.MinX2
my.targetY2 = one2.MinY2
my.shiya = 3
next one2
end if
my.count = my.count +1
makeoneagtsetaroundown(robox,10,universe.二次元空間.マーカー,False)
makeoneagtsetaroundown(robox2,5,universe.二次元空間.マーカー,False)
```

```
if my.count > 50 then
```

```
if countagtset(robox) ==0 then
killagt(my)
end if
```

```
if countagtset(robox2) ==0 then
```

```

turn(180)
end if

end if

for each one in markbox          //通常モード
if (my.targetID == one.ID) then
    my.targetID = one.MinID    //ID 他を更新
    my.targetX = one.MinX
    my.targetY = one.MinY
else
    if my.busy == false then
        my.Direction = GetDirection(my.X, my.Y, my.targetX, my.targetY, Universe.二次元空間)
        //目的のマーカ―へ向かう
        Forward(my.speed/2)
    end if          //目的マーカ―を見つけたら
end if
next one

for each one2 in markbox2      //優先モード
if (my.targetID2 == one2.ID) then
    my.targetID2= one2.MinID2  //ID 他を更新
    my.targetX2= one2.MinX2
    my.targetY2= one2.MinY2
else
    if my.busy == true then
        my.Direction = GetDirection(my.X, my.Y, my.targetX2, my.targetY2, Universe.二次元空間)
        //目的のマーカ―へ向かう
        Forward(my.speed/3)
    end if          //目的マーカ―を見つけたら
end if
next one2

if countagt(universe.二次元空間.優先本部) == 0 then
my.busy = false
end if

if my.busy == false then //通常
    for each one in exitbox    //本部を見つけたらそっちへ向かう
        if CountAgtSet(exitbox) >= 1 Or one.X == my.targetX then
            my.Direction = GetDirection(my.X, my.Y, my.targetX, my.targetY, Universe.二次元空間)
            Forward(my.speed/3)
        end if
    next one
end if

if my.busy == true then //優先
    for each one2 in exitbox2  //本部を見つけたらそっちへ向かう
        if CountAgtSet(exitbox2) >= 1 Or one2.X == my.targetX2 then
            my.Direction = GetDirection(my.X, my.Y, my.targetX2, my.targetY2, Universe.二次元空間)
            Forward(my.speed/3)
        end if
    next one2
end if

If Forward(my.speed) != -1 Then    //動きが止まってしまったら 360° 以内で反転
    Turn(Rnd() * 180 - 30)
End If
マーカ―ルール

```

```

Agt_Init{

    My.color = Color_Yellow
    My.Value = 9999
    My.value2 = 9999
    my.shiya = universe.マーカー視野
    my.shiya2 = universe.マーカー視野
    my.count = 0

}

Agt_Step{

    Dim one As Agt
    dim one2 as agt
    Dim temp As Double
    Dim temp2 As Double
    Dim length As Double
    Dim length2 As Double
    Dim markbox As Agtset
    Dim markbox2 as agtset
    Dim exitbox As Agtset
    dim exitbox2 as agtset
    dim robox as agtset

    my.count = my.count+1

    makeoneagtsetarounddown(robox,2,universe.二次元空間.ロボット,False)

    if my.count >50 then

        if my.Value >150 then
            killagt(my)
        end if
    end if
    MakeOneAgtsetAroundOwn(markbox, my.shiya ,Universe.二次元空間.マーカー, False) //視野範囲に置いてあるマーカーを markbox に格納する
    MakeOneAgtsetAroundOwn(markbox2, my.shiya2 ,Universe.二次元空間.マーカー, False) //視野範囲に置いてあるマーカーを markbox2 に格納する
    MakeOneAgtsetAroundOwn(exitbox,5 ,Universe.二次元空間.本部, False) //視野範囲に置いてある本部を exitbox に格納する
    MakeOneAgtsetAroundOwn(exitbox2,5,universe.二次元空間.優先本部,False)//視野範囲に置いてある優先本部を exitbox2 に格納する

    if (my.Value>=600) and (my.Value!=9999) then //距離計算のループに入ったら
        ClearAgtSet(markbox)
        my.shiya =my.shiya + 1
    end if

    if (my.Value2>=600) and (my.Value2!=9999) then //距離計算のループに入ったら
        ClearAgtSet(markbox2)
        my.shiya2 =my.shiya2 + 1
    end if

    My.Value = 9999 //自分の値をリセット
    My.Value2 = 9999 //自分の値をリセット

    For each one in markbox //markbox に格納されているマーカーに対して繰り返し計算
    if CountAgtSet(markbox)>=1 Then

```



```

My.BMinID = My.MinID
length = Round(sqrt( (My.X - one.X)*(My.X - one.X) + (My.Y - one.Y)*(My.Y - one.Y))) //距離計算
temp = one.Value + length //ノード更新
if (My.Value > temp) Then //最短経路選択
    My.Value = temp //My.Value に temp の値を更新
    My.tMinID = one.ID //最短経路として選択した経路のマーカの ID を tMinID に一時
保管
    My.tMinX = one.X //最短経路として選択した経路のマーカの X 座標を tMinX に一時
保管
    My.tMinY = one.Y //最短経路として選択した経路のマーカの Y 座標を tMinY に一時
保管
end if
end if
next one

For each one2 in markbox2 //markbox2 に格納されているマーカに対して繰り返し計算
if CountAgtSet(markbox2)>=1 Then
    My.BMinID2 = My.MinID2
    length2 = Round(sqrt( (My.X - one2.X)*(My.X - one2.X) + (My.Y - one2.Y)*(My.Y - one2.Y))) //距離
計算
    temp2 = one2.Value2 + length2 //ノード更新
    if (My.Value2 > temp2) Then //最短経路選択
        My.Value2 = temp2 //My.Value2 に temp の値を更新
        My.tMinID2 = one2.ID //最短経路として選択した経路のマーカの ID を tMinID2 に一
時保管
        My.tMinX2 = one2.X //最短経路として選択した経路のマーカの X 座標を tMinX2 に一
時保管
        My.tMinY2 = one2.Y //最短経路として選択した経路のマーカの Y 座標を tMinY2 に一
時保管
    end if
end if
next one2

My.MinID = My.tMinID //最短経路として選択した経路のマーカの ID を MinID に更新
My.MinX = My.tMinX //最短経路として選択した経路のマーカの X 座標を MinX に更新
My.MinY = My.tMinY //最短経路として選択した経路のマーカの Y 座標を MinY に更新

My.MinID2 = My.tMinID2 //最短経路として選択した経路のマーカの ID を MinID2 に更新
My.MinX2 = My.tMinX2 //最短経路として選択した経路のマーカの X 座標を MinX2 に更
新
My.MinY2 = My.tMinY2 //最短経路として選択した経路のマーカの Y 座標を MinY2 に更
新

For each one in exitbox //exitbox に格納されている本部に対して繰り返し計算
if CountAgtSet(exitbox)>=1 Then
    My.BMinID = My.MinID
    length = Round(sqrt( (My.X - one.X)*(My.X - one.X) + (My.Y - one.Y)*(My.Y - one.Y))) //距離計算
    temp = one.Value + length //ノード更新
    if (My.Value > temp) Then //最短経路選択
        My.Value = temp //My.Value に temp の値を更新
        My.tMinID = one.ID //最短経路として選択した経路のマーカの ID を
tMinID に一時保管
        My.tMinX = one.X //最短経路として選択した経路のマーカの X 座標を tMinX に一時
保管
        My.tMinY = one.Y //最短経路として選択した経路のマーカの Y 座標を tMinY に一時
保管
    end if
end if

```

next one

For each one2 in exitbox2 //exitbox2 に格納されている本部に対して繰り返し計算

if CountAgtSet(exitbox2)>=1 Then

My.BMinID2 = My.MinID2

計算  
length2 = Round(sqrt( (My.X - one2.X)\*(My.X - one2.X) + (My.Y - one2.Y)\*(My.Y - one2.Y))) //距離

temp2 = one2.Value2+ length2 //ノード更新

if (my.Value2 >temp2) Then //最短経路選択

My.Value2 = temp2 //My.Value に temp の値を更新

tMinID に一時保管  
My.tMinID2 = one2.ID //最短経路として選択した経路のマーカの ID を

tMinX に一時保管  
My.tMinX2 = one2.X //最短経路として選択した経路のマーカの X 座標を

tMinY に一時保管  
My.tMinY2 = one2.Y //最短経路として選択した経路のマーカの Y 座標を

end if

end if

next one2

My.MinID = My.tMinID //最短経路として選択した経路のマーカの ID を MinID に更新

My.MinX = My.tMinX //最短経路として選択した経路のマーカの X 座標を MinX に更新

My.MinY = My.tMinY //最短経路として選択した経路のマーカの Y 座標を MinY に更新

My.MinID2 = My.tMinID2 //最短経路として選択した経路のマーカの ID を MinID に更新

My.MinX2 = My.tMinX2 //最短経路として選択した経路のマーカの X 座標を MinX に更新

My.MinY2 = My.tMinY2 //最短経路として選択した経路のマーカの Y 座標を MinY に更新

For each one in markbox

if (One.ID == My.MinID) Then //最短経路選択

If My.Color == Color\_Red Then //最短経路表示

one.color = Color\_Red

AddAgt(My.リンク赤, one)

RemoveAgt(My.リンク,one)

else

AddAgt(My.リンク, one)

End if

else

RemoveAgt(My.リンク,one)

RemoveAgt(My.リンク赤,one)

end If

Next one

For each one2 in markbox2

if (One2.ID == My.MinID2) Then //最短経路選択

If My.Color == Color\_Red Then //最短経路表示

one2.color = Color\_Red

AddAgt(My.リンク赤, one)

RemoveAgt(My.リンク,one)

else

AddAgt(My.リンク, one)

End if

else

RemoveAgt(My.リンク,one)

RemoveAgt(My.リンク赤,one)

end If

Next one

```

For each one in exitbox
  if (One.ID == My.MinID) Then //最短経路選択
    If My.Color == Color_Red Then //最短経路表示
      AddAgt(My.リンク赤, one)
      RemoveAgt(My.リンク,one)
    else
      AddAgt(My.リンク, one)
    End if
  else
    RemoveAgt(My.リンク,one)
    RemoveAgt(My.リンク赤,one)
  end If
Next one

For each one2 in exitbox2
  if (One2.ID == My.MinID2) Then //最短経路選択
    If My.Color == Color_Red Then //最短経路表示
      AddAgt(My.リンク赤, one2)
      RemoveAgt(My.リンク,one2)
    else
      AddAgt(My.リンク, one2)
    End if
  else
    RemoveAgt(My.リンク,one2)
    RemoveAgt(My.リンク赤,one2)
  end If
Next one2

```

役所（がれき発生地点）ルール

```

Agt_Init{
My.value = 0 //出口とする
My.color = Color_Cyan
my.shiya = 4
dim 重機 as object
dim i as integer
dim A as agt
for i = 0 to my.重機数 -1
  A = CreateAgt(universe.二次元空間.ロボット)
  A.x = my.X
  A.y = my.Y
next i
Agt_Step{
Dim one As Agt
Dim markbox As Agtset
Dim agtbox as agtset
MakeOneAgtsetAroundOwn(markbox, my.shiya, Universe.二次元空間.マーカー, False) //視野範囲に置いてあるマーカーをmarkboxに格納する
if CountAgtSet(markbox)==0 then //周囲にマーカーがなかったら
  my.shiya = my.shiya + 3 //視野範囲を広げる
end if
For each one in markbox
if (my.shiya >= 10) and CountAgtSet(markbox)>=1 then //自分の広げた視野範囲とマーカーをリンクさせる
  one.shiya = my.shiya
end if
next one

```

```

if GetCountStep() >15 then
makeoneagtsetaroundn(agtbox,2,universe.二次元空間.ロボット,False)
end if
my.重機数 = CountAgtSet(agtbox)
OpenFileCSV("処理ステップ数.csv", 1, 3)
OpenFileCSV("終了ステップ市町村名.csv", 2, 3)
OpenFileCSV("終了ステップがれき処理量.csv", 3, 3)
OpenFileCSV("400step がれき処理量.csv", 5, 3)
OpenFileCSV("400step 市町村名.csv", 6, 3)
openfilecsv("400step 重機数.csv", 7, 3)
openfilecsv("終了ステップ重機数.csv", 8, 3)
if GetCountStep() >15 then
my.step1 = my.step1+1
my.仕事量 = my.仕事量+countagtset(agtbox)*universe.設定仕事量
if my.仕事量>= my.推定がれき発生量*10000 then
WriteFileCSV(8, my.重機数 true)
killagt(my)
WriteFileCSV(1, (GetCountStep()-15), true)
WriteFileCSV(2, my.number, true)
WriteFileCSV(3, my.仕事量, true)
end if
end if
if my.step1 == 400 then
WriteFileCSV(5, my.仕事量, true)
WriteFileCSV(6, my.number, true)
writefilecsv(7, my.重機数, true)
my.step1 = 0
end if
closefilecsv(1)
closefilecsv(2)
closefilecsv(3)
closefilecsv(5)
closefilecsv(6)
closefilecsv(7)
closefilecsv(8)

universe.合計がれき処理量 = universe.合計がれき処理量+countagtset(agtbox)*universe.設定仕事量

}

```

```

Agt_Init{
My.value = 0 //出口とする
My.color = Color_Cyan
my.shiya = 4
dim 重機 as object
dim i as integer
dim A as agt

for i = 0 to my.重機数 -1
A = CreateAgt(universe.二次元空間.ロボット)
A.x = my.X
A.y = my.Y
next i

```

```

Agt_Step{

Dim one As Agt
Dim markbox As Agtset
Dim agtbox as agtset
MakeOneAgtsetAroundOwn(markbox, my. shiya , Universe. 二次元空間. マーカー, False) //視野範囲に置いてあるマーカーをmarkboxに格納する
if CountAgtSet(markbox)==0 then //周囲にマーカーがなかったら
    my. shiya = my. shiya + 3 //視野範囲を広げる
end if
For each one in markbox
if (my. shiya >= 10) and CountAgtSet(markbox)>=1 then //自分の広げた視野範囲とマーカーをリンクさせる
    one. shiya = my. shiya
end if
next one

if GetCountStep() >15 then
makeoneagtsetaroundown(agtbox, 2, universe. 二次元空間. ロボット, False)
end if

my. 重機数 = CountAgtSet(agtbox)

OpenFileCSV("処理ステップ数.csv", 1, 3)
OpenFileCSV("終了ステップ市町村名.csv", 2, 3)
OpenFileCSV("終了ステップがれき処理量.csv", 3, 3)
OpenFileCSV("400stepがれき処理量.csv", 5, 3)
OpenFileCSV("400step市町村名.csv", 6, 3)
openfilecsv("400step重機数.csv", 7, 3)
openfilecsv("終了ステップ重機数.csv", 8, 3)

if GetCountStep() >15 then
my. step1 = my. step1+1
my. 仕事量 = my. 仕事量+countagtset(agtbox)*universe. 設定仕事量
if my. 仕事量>= my. 推定がれき発生量*10000 then
WriteFileCSV(8, my. 重機数 true)
killagt(my)
WriteFileCSV(1, (GetCountStep()-15) , true)
WriteFileCSV(2, my. number, true)
WriteFileCSV(3, my. 仕事量, true)

end if
end if
if my. step1 == 400 then
WriteFileCSV(5, my. 仕事量, true)
WriteFileCSV(6, my. number, true)
writefilecsv(7, my. 重機数, true)
my. step1 = 0
end if
closefilecsv(1)
closefilecsv(2)
closefilecsv(3)
closefilecsv(5)
closefilecsv(6)
closefilecsv(7)
closefilecsv(8)
universe. 合計がれき処理量 = universe. 合計がれき処理量+countagtset(agtbox)*universe. 設定仕事量

```

}

## 参考文献

- 1) 馬場太一郎:高知県の建設業と住民による地域防災に関する基礎的研究,平成21年度フロンティアプロジェクト修士論文, 2010
- 2) 金子雄一郎・兵藤哲郎:東日本大震災による交通システムの機能障害の発生状況と復旧について,「緊急特集」東日本大震災,pp.83-85 2011.
- 3) 財務省:各論 3. 公共事業,公共事業関連予算の推移, 2011.
- 4) 国土交通省:国土交通白書,図表 102, 2011.
- 5) 静岡県:第3次被害想定結果:地震対策資料 No182-201, 2001.
- 6) 宮島宇・堀宗朗・小国健二:多様な群集の雑然とした状況を想定した地震時避難行動シミュレーション,土木学会地震工学論文集, pp765-772, 2007.
- 7) 笹岡早姫:災害現場における最短経路探索システムの構築宮崎隆穂,第11回 MAS コンペティション,アブストラクト, pp.1-2, 2011.
- 8) 前地一輝:DIGへの適用を目的としたマルチエージェント避難行動シミュレータの試験的開発,第10回 MAS コンペティション,アブストラクト, 2010.
- 9) 野澤征司・渡辺公次郎・近藤光男:マルチエージェントシステムを用いた歴史的市街地における津波避難シミュレーションモデルの構築,土木学会四国支部技術研究発表会講演概要集, pp. 230-231, 2005.
- 10) 村木雄二:マルチエージェントモデルを用いた広域災害避難シミュレーションにおける情報伝達の有効性,情報処理学会研究報告 2004 号, pp. 69-72, 2004.
- 11) 静岡県庁交通基盤部:重機・オペレーター保有数の集計, 2011
- 12) 社団法人中部経済連合会:大震災に備えた震災がれき処理について～災害に強く,環境にやさしい中部圏を目指して～, pp5, 2008.
- 13) 国土交通省:土木工事標準積算書, 2011
- 14) 国土交通省:平成22年度 道路交通センサス一般交通量調査結果の概要について, pp1, 2011.

