

平成 21 年度
博士論文

多品種を考慮したロジスティクスネットワーク設計
問題の数理的解法に関する研究

流通経済大学大学院 物流情報学研究科
片山 直登

あらまし

1980年代以降の貿易摩擦や円高の影響を受けて、多くの企業は生産拠点を海外に移転し、さらに中国や東南アジア諸国の経済発展とともに市場の中心も海外にシフトしてきている。生産拠点や市場先が海外に展開するにつれて、ロジスティクスは地球規模で調達、生産、輸送、販売を行うグローバルロジスティクスに進展してきている。このように、ロジスティクスはその規模を拡大し、ロジスティクスネットワークを最適に設計することが企業の最重要課題となっている。

今日まで、多くの企業が生産拠点や物流拠点など拠点単独でのコスト削減や効率化に取り組んできた。しかし、アメリカ発の不況の急激な進行により、ロジスティクスネットワークの構造全体を通じた大幅な再編を行うことが急務となっている。たとえば、国内や海外における生産拠点の開設、閉鎖および移転、物流拠点の統合や分散、輸送手段の選択や変更などである。

多数の生産拠点、多数の物流拠点、複数の輸送手段、そして多数の品種など数多くの要因を考慮しなければならないため、適切なロジスティクスネットワークを設計することは、非常に困難なものとなっている。このようなロジスティクスネットワークを設計するための数理計画モデルとして、ロジスティクスネットワーク設計問題がある。今日まで、多くのロジスティクスネットワーク設計問題のモデルと解法が開発され、数多くの企業に適用されてきており、大きな成果を挙げてきている。しかし、近年では、グローバル化とともにロジスティクスネットワークの規模がさらに拡大してきているために、大規模なロジスティクスネットワークに適用でき、かつ高速で精度の高いモデルと解法が開発が望まれている。

多品種を考慮したロジスティクスネットワーク設計問題は、ロジスティクスネットワークの形状と多品種のモノを輸送する経路などを求める問題である。この問題には、ロジスティクスネットワークのみならず、交通ネットワークや通信ネットワークまでの幅広い応用分野が存在する。本論文では、多品種を考慮したロジスティクスネットワーク設計問題の中で、容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、および区分的線形費用をもつネットワーク設計問題を対象とする。これらの問題は、ロジスティクスネットワークを設計し、構築する際に発生する問題を抽象化したものであり、現実において考慮すべき様々な条件の中で基本的かつ重要な項目を考慮してモデル化したものである。本論文で取り扱う問題は、数理面から最適に解くことが困難である *NP* 困難な問題のグループに属している。このため、規模の小さな問題に対しては厳密な最適解を求めることは可能であるが、実用規模の問題に対する一般的、効率的な最適解法は存在していない。

本論文で対象とするロジスティクスネットワーク設計問題は、必ずしも現実のロジスティクスにおけるネットワーク設計における条件をすべて考慮している訳ではないが、ネットワーク設計に関する根幹的な条件を考慮している。現実のロジスティクスネット

ワーク設計問題に対しては、本論文で対象とする基本的な問題の解をもとにロジスティクスネットワーク全体の構造を決定し、このネットワークの構造を基本として、さらなる詳細な条件を考慮した下位の問題に分解し、これらを解くことによって全体を解くことができるようになる。またその一方で、基本的な問題に対して有効な解法は、より複雑な条件をもつ現実の問題に対する解法への組み込みや拡張、応用ができる可能性がある。このように、基本的な問題に対する解法は現実的なロジスティクスネットワーク設計問題の解法の基礎または一部となり、現実の問題に対する解法の開発と解決に有益なものとなる。

今日まで、多品種を考慮したロジスティクスネットワーク設計問題に対して、数多くの研究が行われてきた。これらは、問題構造の解析的な研究、より良い解を算出するための近似解法の研究、緩和問題を解くことにより良い下界値を算出するための解法の研究、実際の企業における事例問題に焦点を当てた事例的な研究などである。本論文では、これらの中でより良い解を算出する解法、より短時間で解を算出するための解法の開発、および緩和問題に対する解法の開発を主な対象としている。

より良い解とは、最適な値により近い解や誤差の小さい解を意味する。ロジスティクスネットワークを構築する際には多額の費用が必要となるため、たとえ数パーセントの改善であっても、結果として費用削減のための大きな貢献と成り得る。より短時間で解を算出できることは、より大きな問題を解くことができる可能性があることも意味する。より短時間で解を算出できる解法であれば、安価なコンピュータ環境下であっても、より大きな規模の現実の問題を妥当な時間で解くことができる可能性が生じる。また、より複雑なネットワーク設計問題には、安定して実行可能な解を提供できていない問題が存在している。このような複雑な問題に対しては、常に適切な計算時間内で実行可能解を算出できる解法の開発が必要である。

容量制約のないネットワーク設計問題に対しては、高速な近似解法や精度の高い解を算出する近似解法が開発されている。高速な近似解法は算出する解の精度が悪い場合があり、その誤差は問題の規模にしたがって大きくなる傾向にある。高額なネットワーク設計の費用を考えると、その誤差による損失は大きなものとなる。また、従来の精度の高い解を算出する近似解法は大きな計算時間を必要とするため、実用規模の問題の解を実用的な時間では算出できないという状況も生じる可能性がある。本論文では、このような点を踏まえ、従来の精度の高い解を算出する近似解法と同程度の誤差で、かつ計算時間を大幅に抑えることのできる近似解法を開発している。またその一方で、計算時間はかかるものの、従来の精度の高い解を算出する近似解法よりもさらに精度の高い解を算出する近似解法を開発している。

近似解法によって得た近似解の良さを保証するためには、下界値とよばれる目的とするものの下限値が必要になる。近似解の良さをより正確に表すためには、より良い下界値を求めることが必要である。また、上界値により近い下界値をもつ解からは、良い近似解を導出できる可能性がある。本論文では、容量制約のないネットワーク設計問題に対して、従来の下界値を求める解法よりも計算時間は大きくなるものの、より良い下界値を求める解法を開発している。

容量制約のあるネットワーク設計問題に対しては、メタ解法とよばれる近似解法が数多く開発されている。容量制約をもつ設計問題は、容量制約のない問題に比べて規模がそれほど大きくない問題であっても、高速に良い解を算出する近似解法が開発されていない状況である。本論文では、容量制約のある設計問題に対して、従来の研究よりも平均的に精度が高く、かつ安価なパソコンを用いても高速に解くことができる解法を開発している。

区分的な線形費用をもつネットワーク設計問題は、より現実的な費用を考慮した問題である。容量制約のあるネットワーク設計問題と比べてもさらに複雑な問題であり、それほど規模が小さくない問題であっても適切な解を求めることが困難な場合がある。また、近年研究が始まったばかりであることもあって、いまだ一般的に普及している解法が提供されていない状況にある。本論文では、容量制約のある問題に適用した近似解法を区分的な費用をもつ設計問題へ適用できるように拡張することによって、ある程度の規模までの問題を解くことができる解法を開発している。

本論文で得られた成果をまとめると次のようになる。

- ・容量制約のないネットワーク設計問題に対して、大規模な問題に対しても適用が可能である高速な新しい近似解法を開発している。さらに、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発している。
- ・容量制約をもつネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発している。
- ・区分的線形費用をもつネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発している。

各章毎の内容を示す。

第1章では、研究の背景、研究の目的、問題の提示および論文の構成を示している。問題の提示では、本論文に關係する用語の定義、ロジスティクスネットワーク設計問題の定義を示し、その分類を解説とロジスティクスにおける応用問題の例を解説している。

第2章では、従来の研究の整理を行い、従来の研究における本論文の各章の位置づけを示している。従来の研究では、関連する問題を容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、区分的線形費用をもつネットワーク設計問題、応用問題、およびサーベイに分類し、解説している。

第3章では、容量制約のないネットワーク設計問題を取り扱い、この問題に対する強い妥当不等式といくつかの近似解法を示し、数値実験により提案した解法の有効性を示している。はじめに、新たな強い妥当不等式である非順流不等式を示し、非逆流不等式と非順流不等式を制約式として含むアークフローによる定式化とパスフローによる定式化を示している。次に、Minoux法におけるアーク削除時の目的関数の評価に厳密性を加える改良を行ったMinoux法の改良法を示している。さらに、パスフロー変数を用いた定式化に対する線形緩和問題を列生成法と行生成法を用いて解き、この解から近似解を導出する線形緩和法を示している。最後に、数値実験により提案した解法の有効性を示している。Minoux法の改良法は、バックワード法と同程度の精度の解を求めることができ、かつバックワード法に比べて大幅に計算時間を短縮することができる。線形緩和法は、

多くの計算時間を必要とするが，従来の解法よりも優れた上界値を求めることができている．

第4章では，第3章と同じ問題を取り扱い，この問題に対していくつかの Lagrange 緩和法を示し，数値実験により提案した解法の有効性を示している．はじめに，品種と需要に関する強制制約を含む定式化，非逆流不等式と連結条件を含む定式化，非逆流不等式と非順流不等式を含む定式化を示している．次に，第一の定式化に対して，フロー保存式を Lagrange 緩和した問題がアーク毎の問題に分割できることを示し，この緩和問題がデザイン変数のみを含む問題として表現できることを示している．続いて，第二の定式化に対して，フロー保存式を Lagrange 緩和した問題がアーク毎，始点毎の問題に分割することができることを示し，分割された問題が最小木問題に帰着されることを示している．さらに，第三の定式化に対して，非逆流と非順流不等式を Lagrange 緩和した問題が最小木問題に帰着されることを示し，Lagrange 乗数の数を制御するための Lagrange 乗数値の設定方法を示している．また，Lagrange 緩和解をもとにした近似解法を示している．最後に，数値実験により提案した解法の有効性を示している．いずれの Lagrange 緩和法も，多くの計算時間を必要とするが，従来の解法よりも良い下界値を算出することができる．

第5章では，容量制約のあるネットワーク設計問題を取り扱い，容量スケールリング法を用いた近似解法を提案し，数値実験により提案した解法の有効性を示している．はじめに，アークフローを用いた弱い定式化と強い定式化，およびパスフローを用いた強い定式化を示している．次に，アークフローを用いた定式化に対する容量スケールリング法を示し，得られた解を用いた近似解法を示している．続いて，パスフローを用いた定式化に対して列生成法と行生成法を用いた容量スケールリング法を示し，得られた解をもとにした限定した分枝限定法を示している．最後に，数値実験により提案した解法の有効性を示している．強い定式化に対する容量スケールリング法は，従来の解法に比べて多くの問題で良い解を求めることができている．また，パスフローを用いた定式化に対する容量スケールリング法は，列生成法と行生成法により大幅に計算時間を短縮することができる．

第6章では，区分的線形費用をもつネットワーク設計問題を取り扱い，アークフローによる定式化とパスフローによる定式化に対する容量スケールリング法を用いた近似解法を示し，数値実験により提案した解法の有効性を示している．はじめに，アークフローを用いた定式化に対して，容量スケールリング法とその解をもとにした分枝限定法を示している．次に，パスフローを用いた定式化に対して，列生成法と行生成法を用いた容量スケールリング法を示し，得られた解をもとにした分枝限定法を示している．最後に，数値実験により提案した解法の有効性を示している．従来の解法が実行可能解を算出できない場合があるのに対して，すべての問題で実行可能解を算出することができる．アークフローを用いた容量スケールリング法は，品種数の大きな問題以外の問題では従来の解法よりも良い解を得ることができる．また，パスフローフローを用いた容量スケールリング法は，計算時間を大幅に短縮することができる．

第7章では，本論文のまとめと今後の課題を記述している．

目次

第 1 章	序論	1
1.1	研究の背景	1
1.2	研究の目的	2
1.3	問題の提示	3
1.4	論文の構成	14
第 2 章	従来の研究	17
2.1	従来の研究	17
2.2	従来の研究と本論文の各章の位置づけ	31
第 3 章	容量制約のないネットワーク設計問題の近似解法	35
3.1	本章の概要	35
3.2	問題の定式化	36
3.3	強い妥当不等式	40
3.4	Minoux 法の改良法	42
3.5	線形緩和法	46
3.6	数値実験	57
3.7	本章のまとめ	67
第 4 章	容量制約のないネットワーク設計問題の Lagrange 緩和法	71
4.1	本章の概要	71
4.2	問題の定式化	71
4.3	基本的な定式化に対する Lagrange 緩和法	75
4.4	強い定式化に対する Lagrange 緩和法	79
4.5	拡張した定式化に対する Lagrange 緩和法	83
4.6	Lagrange ヒューリスティック	90
4.7	数値実験	93
4.8	本章のまとめ	100
第 5 章	容量制約をもつネットワーク設計問題の近似解法	103

5.1	本章の概要	103
5.2	問題の定式化	104
5.3	アークフローを用いた定式化に対する容量スケーリング法	113
5.4	パスフローを用いた定式化に対する容量スケーリング法	119
5.5	数値実験	125
5.6	本章のまとめ	136
第 6 章	区分的線形費用をもつネットワーク設計問題の近似解法	143
6.1	本章の概要	143
6.2	問題の定式化	143
6.3	アークフローを用いた定式化に対する容量スケーリング法	154
6.4	パスフローを用いた定式化に対する容量スケーリング法	164
6.5	数値実験	172
6.6	本章のまとめ	190
第 7 章	結論	193
7.1	まとめ	193
7.2	今後の課題	198
参考文献		203
付録 A		209
付録 B		219
研究業績		231
謝辞		239

目次

1.1	ネットワーク設計問題	7
1.2	<i>UND</i> におけるフロー量と費用の関係	9
1.3	<i>CND</i> におけるフロー量と費用の関係	10
1.4	<i>PND</i> におけるフロー量と費用の関係	11
1.5	ストラテジックレベルのロジスティクスネットワーク設計問題	12
1.6	タクティカルレベルのロジスティクスネットワーク設計問題	13
1.7	オペレーショナルレベルのロジスティクスネットワーク設計問題	14
3.1	非集約化した強制制約式	39
3.2	非逆流不等式	41
3.3	非順流不等式	42
3.4	目的関数値の減少量	43
5.1	線形関数とギャップ	114
5.2	デザイン変数とフロー変数の関係	115
5.3	アーク容量の変更	116
6.1	区分的線形関数	146
6.2	多重ダミーアーク	152
6.3	線形関数とギャップ	155
6.4	区分変数とフロー変数の関係	156
6.5	ダミーアークとフロー量	160
6.6	区分フロー量と区分変数値	161

表目次

1.1	ネットワーク設計問題の分類	8
2.1	従来の研究と本論文の各章の位置づけ	33
2.2	多品種問題に対する従来の研究と本論文の各章の位置づけ	34
3.1	問題の属性 1	59
3.2	問題の属性 2	59
3.3	Minoux の改良法：上界値の誤差の比較	60
3.4	Minoux の改良法：計算時間の比較	61
3.5	Minoux の改良法：大規模問題：上界値の比較	62
3.6	Minoux の改良法：大規模問題：計算時間の比較	63
3.7	線形緩和法：上界値の誤差の比較	64
3.8	線形緩和法：計算時間の比較	65
4.1	問題の属性 1	94
4.2	問題の属性 2	94
4.3	下界値の誤差の比較	95
4.4	下界値の計算時間の比較	96
4.5	上界値の誤差の比較	97
4.6	上界値の計算時間の比較	98
5.1	C 問題:属性	126
5.2	C 問題:解法別平均誤差 従来の解法	127
5.3	C 問題:解法別平均誤差 容量スケーリング	127
5.4	C 問題:性質別平均誤差	127
5.5	C 問題:性質別最良値数	128
5.6	パスフローによる定式化 C 問題:規模別平均誤差	128
5.7	C 問題:規模別最良値数	129
5.8	C 問題:個別結果 1	130
5.9	C 問題:個別結果 2	131

5.10	C 問題:性質別平均計算時間	132
5.11	C 問題:規模別平均計算時間	132
5.12	C 問題:個別計算時間 1	133
5.13	アークフローによる定式化 C 問題:計算時間 2	134
5.14	R 問題:属性 1	136
5.15	R 問題:属性 2	136
5.16	R 問題:性質別平均誤差 1 ^{*1}	137
5.17	R 問題:性質別平均誤差 2 ^{*2}	137
5.18	R 問題:性質別最良値数	138
5.19	R 問題:規模別平均誤差 1 ^{*1}	139
5.20	R 問題:規模別平均誤差 2 ^{*2}	140
5.21	R 問題:規模別最良値数	141
5.22	R 問題:性質別平均計算時間	141
5.23	R 問題:規模別平均計算時間	142
6.1	C 問題の属性	174
6.2	C 問題:実行可能解数	174
6.3	C 問題:解法別平均誤差 ^{*1}	175
6.4	C 問題:性質別平均誤差 ^{*1}	175
6.5	C 問題:規模別平均誤差 ^{*1}	176
6.6	C 問題:上界値の比較	177
6.7	C 問題:上界値の誤差の比較	178
6.8	C 問題:区分スケージングの繰返し回数	179
6.9	C 問題:性質別計算時間 ^{*1}	179
6.10	C 問題:規模別計算時間 ^{*1}	180
6.11	C 問題:計算時間	181
6.12	R 問題:属性 1	182
6.13	R 問題:属性 2	182
6.14	R 問題:性質別平均誤差 ^{*1}	183
6.15	R 問題:規模別平均誤差	184
6.16	R 問題:性質別最適解数	185
6.17	R 問題:規模別最適解数	186
6.18	R 問題:性質別平均計算時間 ^{*1}	187
6.19	R 問題:規模別平均計算時間 ^{*1}	188
A.1	CND R 問題:個別結果 1	210
A.2	CND R 問題:個別結果 2	211
A.3	CND R 問題:個別結果 3	212

A.4	<i>CND</i>	R 問題:個別結果 4	213
A.5	<i>CND</i>	R 問題:個別結果 5	214
A.6	<i>CND</i>	R 問題:個別結果 6	215
A.7	<i>CND</i>	R 問題:個別計算時間 1	216
A.8	<i>CND</i>	R 問題:個別計算時 2	217
A.9	<i>CND</i>	R 問題:個別計算時間 3	218
B.1	<i>PND</i>	R 問題:個別結果 1	220
B.2	<i>PND</i>	R 問題:個別結果 2	221
B.3	<i>PND</i>	R 問題:個別結果 3	222
B.4	<i>PND</i>	R 問題:個別結果 4	223
B.5	<i>PND</i>	R 問題:個別結果 5	224
B.6	<i>PND</i>	R 問題:個別計算時間 1	225
B.7	<i>PND</i>	R 問題:個別計算時間 2	226
B.8	<i>PND</i>	R 問題:個別計算時間 3	227
B.9	<i>PND</i>	R 問題:個別計算時間 4	228
B.10	<i>PND</i>	R 問題:個別計算時間 5	229

第1章

序論

1.1 研究の背景

1980年代以降の貿易摩擦や円高の影響を受けて、多くの企業は生産拠点を海外に移転し、さらに中国や東南アジア諸国の経済発展とともに市場の中心も海外にシフトしてきている。生産拠点や市場先が海外に展開するにつれて、ロジスティクスは地球規模で調達、生産、輸送、販売を行うグローバルロジスティクスに進展してきている。このように、ロジスティクスはその規模を拡大し、ロジスティクスネットワークを最適に設計することが企業の最重要課題となっている。

今日まで、多くの企業が生産拠点や物流拠点など拠点単独でのコスト削減や効率化に取り組んできた。しかし、アメリカ発の不況の急激な進行により、ロジスティクスネットワークの構造全体を通じた大幅な再編を行うことが急務となっている。たとえば、国内や海外における生産拠点の開設、閉鎖および移転、物流拠点の統合や分散、輸送手段の選択や変更などである。

多数の生産拠点、多数の物流拠点、複数の輸送手段、そして多数の品種など数多くの要因を考慮しなければならないため、適切なロジスティクスネットワークを設計することは、非常に困難なものとなっている。このようなロジスティクスネットワークを設計するための数理計画モデルとして、ロジスティクスネットワーク設計問題がある。今日まで、多くのロジスティクスネットワーク設計問題のモデルと解法が開発され、数多くの企業に適用されてきており、大きな成果を挙げている。しかし、近年では、グローバル化とともにロジスティクスネットワークの規模がさらに拡大してきているために、大規模なロジスティクスネットワークに適用でき、かつ高速で精度の高いモデルと解法が開発が望まれている。

多品種を考慮したロジスティクスネットワーク設計問題は、ロジスティクスネットワークの形状と多品種のモノを輸送する経路などを求める問題である。この問題には、ロジスティクスネットワークのみならず、交通ネットワークや通信ネットワークまでの幅広い応用分野が存在する。本論文では、多品種を考慮したロジスティクスネットワーク設計問題の中で、容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、

および区分的線形費用をもつネットワーク設計問題を対象とする。これらの問題は、ロジスティクスネットワークを設計し、構築する際に発生する問題を抽象化したものであり、現実において考慮すべき様々な条件の中で基本的で重要な項目を考慮してモデル化したものである。本論文で取り扱う問題は、数理面から最適に解くことが困難である NP 困難な問題のグループに属している。このため、規模の小さな問題に対しては厳密な最適解を求めることは可能であるが、実用規模の問題に対する一般的、効率的な最適解法は存在していない。

ロジスティクスネットワーク設計問題は、必ずしも現実のロジスティクスにおけるネットワーク設計における条件をすべて考慮している訳ではないが、ネットワーク設計に関する根幹的な条件を考慮している。現実のロジスティクスネットワーク設計問題に対しては、基本的な問題の解をもとにロジスティクスネットワーク全体の構造を決定し、このネットワークの構造を基本として、さらなる詳細な条件を考慮した下位の問題に分解し、これらを解くことによって全体を解くことができるようになる。またその一方で、基本的な問題に対して有効な解法は、より複雑な条件をもつ現実の問題に対する解法への組み込みや拡張、応用ができる可能性がある。このように、基本的な問題に対する解法は、現実的なロジスティクスネットワーク設計問題の解法の基礎または一部となり、現実の問題に対する解法の開発と解決に有益なものとなる。

今日まで、多品種を考慮したロジスティクスネットワーク設計問題に対しては数多くの研究が行われてきた。これらは、問題構造の解析的な研究、より良い解を算出するための近似解法の研究、緩和問題を解くことにより良い下界値を算出するための解法の研究、実際の企業における事例問題に焦点を当てた事例的な研究などである。より良い解とは、最適な値により近い解や誤差の小さい解を意味する。ロジスティクスネットワークを構築するには多額の費用が必要となるため、たとえ数パーセントの改善であっても、結果として費用削減のための大きな貢献と成り得る。数理計画の問題において、その規模によって最速のコンピュータを用いても数年、数万年かかるであろう問題はざらにある。より短時間で解を算出できることは、より大きな問題を解くことができる可能性を意味する。より短時間で解を算出できる解法であれば、安価なコンピュータ環境下であっても、より大きな規模の現実の問題を妥当な時間で解けることもあり得る。また、より複雑なネットワーク設計問題には、安定して実行可能な解を提供できていない問題が存在している。

このようなことから、少しでも誤差の小さい解を算出できる解法、計算時間を短縮できる解法や、常に適切な計算時間内に実行可能解を算出できる解法の開発が望まれている。

1.2 研究の目的

本論文では、多品種を考慮したネットワーク設計問題の中で、容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、および区分的線形費用をもつネットワーク設計問題を対象とする。これらの問題は、ネットワークを設計し、構築する際に発生する問題を抽象化したものであり、現実において考慮すべき様々な条件の中で基

本的で重要な項目を考慮してモデル化したものである。本論文で取り扱う問題は、数理面から最適に解くことが困難である NP 困難な問題のグループに属している。このため、規模の小さな問題に対しては厳密な最適解を求めることは可能であるが、実用規模の問題に対する一般的、効率的な最適解法は存在していない。このため、これらの問題に対する実用的な解を求める近似解法を開発することが課題となる。

今日まで、多品種を考慮したネットワーク設計問題に対する数多くの研究が行われおり、多くの成果が上げられている。しかしながら、ロジスティクスネットワークの規模の拡大や問題の複雑化により、大規模な問題に適用できない、解の精度が劣り多額の費用の損失が生じる、複雑なネットワーク設計問題に対して安定して解を提供できないといったことが生じる可能性があり、従来の研究には改良の余地が残されている。

本論文では、従来の研究を整理し、問題の性質を分析するとともに、より規模の大きな問題に対して適用できる解法、より精度の高い解を算出できる解法、複雑な問題に対して安定して解を提供できる解法などの新しい近似解法を提案する。

本論文の目的をまとめると次のようになる。

- 容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、および区分的線形費用をもつネットワーク設計問題に関して、従来の研究を整理する。
- 容量制約のないネットワーク設計問題に対して、強い妥当不等式を提示する。
- 容量制約のないネットワーク設計問題に対して、大規模な問題に対しても適用が可能である高速な新しい近似解法を開発する。
- 容量制約のないネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発する。
- 容量制約をもつネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発する。
- 区分的線形費用をもつネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発する。

1.3 問題の提示

この節では、ネットワークと最適化問題の項に分けて、本論文で使用する用語を説明する。続いて、本論文で取り扱うネットワーク設計問題についての概要を説明する。最後に、ロジスティクスにおけるネットワーク設計問題の応用について解説する。

1.3.1 用語定義

ネットワーク

ネットワークは、モノの移動を伴う点と線によって構成される図形である。ネットワーク上では、決められた二点間をモノが移動する。モノは、例えば荷物、商品などを表しており、これらを抽象化したものである。ネットワーク上の点をノードとよぶ。ノードはモノの移動が発生、集中するまたは中継される場所や地点を表しており、例えば工場、倉庫、配送施設、消費地、顧客などを抽象化したものである。ネットワーク上で点と点をつなぐ線をアークとよび、アーク上をモノが移動する。アークは例えば輸送便、輸送路、道路区間などを抽象化したものである。また、モデルによっては、アークが生産・在庫といった工程や時間的経過を表す場合もある。

モノが出発するノードを始点、到着するノードを終点とよぶ。始点を同一ノード、終点を同一ノードとするモノの集まりを同一の品種と定義し、一つの単位として取り扱う。一般的には、品種はモノの違いを表すが、本論文では始点・終点の組合せの違いを表す。ある品種の始点と終点間の移動すべきモノまたはその量を需要とよぶ。同一のネットワーク上で一つの品種を取り扱うモデルを一品種モデル、複数の品種を取り扱うモデルを多品種モデルとよぶ。本論文では、多品種モデルに対するネットワーク設計問題を取り扱う。

アークには二種類の費用が与えられている。デザイン費用は、アークを選択するときに発生する固定的な費用である。これは、路線便を開設する費用や維持するために必要な固定的な費用に対応する。フロー費用は、アーク上をモノが移動するときに移動量に関連して発生する変動的な費用である。これは、輸送量に依存する輸送費用に対応する。また、アークにはアーク容量が与えられる場合がある。アーク容量は、単位期間にアーク上をモノが移動できる量や処理できる量の上限を表す。これは、路線便の輸送能力や便数に対応する。

アークには方向性をもつものともたないものがあり、アーク上で移動できる方向が決まっているアークを向きをもつアーク、両方向の移動ができるアークを向きをもたないアークとよぶ。向きをもつアークは復路を考慮する必要のない他社便利用の場合、向きをもたないアークは復路も考慮する自社便利用の場合などに相当する。アークの両端のノードをアークの端点とよぶ。アークが向きをもつ場合、アークが出るノードをアークの始点、入るノードをアークの終点とよぶ。二つのノード間に複数のアークがあるものを多重アークとよぶ。多重アークは、複数の輸送手段から選択できる場合に相当する。

二つのノード間をつなぐノードとアークの集合をパスまたは経路とよぶ。パスは、モノが移動する道筋であり、モノの輸送経路を表す。ネットワークにおいて、すべてのノード間にパスが存在することをネットワークが連結するとよぶ。アークに関連する費用をアークの長さと考えた場合、始点と終点間のパスに含まれるアークの長さの合計を始点・終点間の距離とよぶ。ある品種について、距離が最小であるパスを最短パス、その距離を最短距離とよぶ。始点・終点間をモノが移動するとき、そのパスと移動する量を合わせてフ

ロー，移動する量をフロー量とよぶ。

最適化問題

最大化または最小化したい関数を目的関数とよび，変数が満足しなければならない式を制約式とよぶ。目的関数と制約式で構成される問題を最適化問題とよぶ。以下では，目的関数値を最小化する問題を想定して記述を行う。

目的関数と制約式を用いて，問題を数式によって記述したものを定式化とよぶ。同一の問題であっても，複数種類の定式化で記述できることが多い。制約式を満足する変数値を実行可能解とよび，実行可能解を含む領域を実行可能領域とよぶ。また，制約式を満足し，かつ目的関数値が最小となるような変数値を最適解，その目的関数値を最適値とよぶ。

変数がすべて連続数であり，目的関数と制約式が共に線形関数で表される問題を線形計画問題とよぶ。また，0 または 1 で表される離散的な変数を含む問題を組合せ最適化問題とよぶ。一般的に，線形計画問題は単体法などの解法により大規模な問題であっても効率的に最適解を求めることができる。組合せ最適化問題の内，理論的に最適に解くのが難しいとされている問題を *NP* 困難な問題とよぶ。*NP* 困難な問題に対しては，一般的，効率的な解法は見つかっていない。そのため，有効な近似解法が必要となる。本論文で対象とするネットワーク設計問題は，すべて *NP* 困難な問題である。

制約式を満足し，かつ目的関数値が最適値に近いであろう解を近似解とよび，近似解を求める解法を近似解法とよぶ。近似解における目的関数値は最適値以上となることから，この値を上界値とよぶ。また，現時点までに得られた最良の目的関数値や，解法によって得られた最良の目的関数値を上界値とよぶことがある。上界値の良さは，下界値または最適値との差である誤差により評価することができる。

一部の制約式を緩和した問題を緩和問題とよぶ。変数の 0-1 条件を表す式を 0 から 1 の連続数である連続条件の式に緩和することを線形緩和とよぶ。また，制約式に乗数を掛け，目的関数に組み込む緩和を Lagrange 緩和とよぶ。緩和問題の実行可能解を緩和解とよぶ。緩和問題の最適値は元の問題の最適値以下となり，最適値の下限を表すことになるため，緩和問題の最適値を下界値とよぶ。下界値は，近似解法による上界値の誤差の保証や分枝限定法などに用いられる。最適値に近い下界値は上界値の誤差をより正確に記述することができる。また，良い緩和解から良い近似解が導出できることが多い。このようなことから，最適値により近い下界値を求めることが重要である。

近似解法には，貪欲法，局所探索法，各種のメタ解法や緩和法などがある。貪欲解法は，何らかの基準によって変数値を順次，確定していき，試行錯誤を行わない方法であり，一般的に計算量を抑えることができる。局所探索法は，現在の変数値の一部を変更する近傍探索を行い，目的関数値が減少する近傍解への移動を繰り返す方法である。メタ解法は，局所探索法などに確率的な要素や禁止事項などを組み込んだ方法であり，タブー探索法，アニーリング法，遺伝的アルゴリズム，パス再結合法やスケールリング法などさまざまな解法がある。また，緩和法は，緩和問題を解き，その解を改善して実行可能解を求める方法であり，双対ヒューリスティックや Lagrange ヒューリスティックなどがある。

定式化において、すべての実行可能解を満足する不等式や最適解を満足する不等式を妥当不等式とよぶ。このうち、線形緩和問題の一部の実行可能解を満足しない式を有効な妥当不等式とよぶ。有効な妥当不等式を制約式として追加すると、実行可能領域が狭まる。このため、妥当不等式を制約式として追加した問題から得られる下界値は、追加する前の問題から得られる下界値よりも増加する可能性がある。このため、有効な妥当不等式を求めることが重要となる。より実行可能領域を狭める妥当不等式を強い妥当不等式とよび、強い妥当不等式を含む定式化を強い定式化とよぶ。強い妥当不等式の数、一般的に膨大なものとなるため、逐次、現在の緩和問題における有効な妥当不等式のみを生成することが重要となる。

制約式を適当な集合の要素毎の式に分解することを非集約化とよび、逆の操作を集約化とよぶ。一般的に、非集約化した妥当不等式は、集約化した妥当不等式よりも強い妥当不等式となる。

問題の定式化において、 $|\cdot|$ は集合の要素数を表し、例えば $|N|$ は集合 N の要素数を表す。また、変数の太文字はベクトル表現であり、例えば $x_{ij}(i, j \in N)$ のベクトル表現は $x = (x_{ij})$ となる。アーク上のフローを表す変数をアークフロー変数、パス上のフローを表す変数をパスフロー変数、ネットワークにアークを選択するか否かを表す 0-1 変数をデザイン変数とよぶ。

定数項と 0-1 であるデザイン変数の積によって、フロー量などの変数値の上限を抑える制約式を強制制約式とよぶ。デザイン変数が 0 である場合に強制的にフロー量を 0 にさせることから、このようによばれる。

計算量とは、ある大きさのデータに対して、特定の解法により解を算出するためにどれだけの計算時間あるいは記憶領域を必要とするかを表す。計算時間に関する量を時間計算量、記憶領域に関する量を領域計算量とよぶ。単に計算量と記述した場合、本論文では時間計算量を意味する。 $O(\cdot)$ は計算量を表す O 表記法である。例えば、計算量を $f(|N|)$ 、 c を正の定数とすると、 $0 < f(|N|) < c|N|^2$ のように計算量の上限が $c|N|^2$ で抑えられるとき、解法の計算量を $O(|N|^2)$ と表現する。

1.3.2 ネットワーク設計問題

ネットワーク設計問題は、輸送やロジスティクスなどの領域におけるネットワークを設計する問題である。ネットワーク設計問題の概念図を図 1.1 に示す。ネットワークを設計する際に決めるべき最も基本的な事項は、どの点とどの点をアークでつなぐかというネットワークの構成と、モノをどのパス上でいくつ移動するかというフローの決定である。ネットワーク設計問題は、ノード集合、アーク集合および品種集合とその需要が与えられたときに、デザイン費用やフロー費用を考慮して、アーク集合から使用するアークを選択してネットワークを構成し、このネットワーク上で需要が移動するパスと量であるフローを求める問題である。

条件や前提の組合せによって、ネットワーク設計問題はさまざまな問題に分類される。

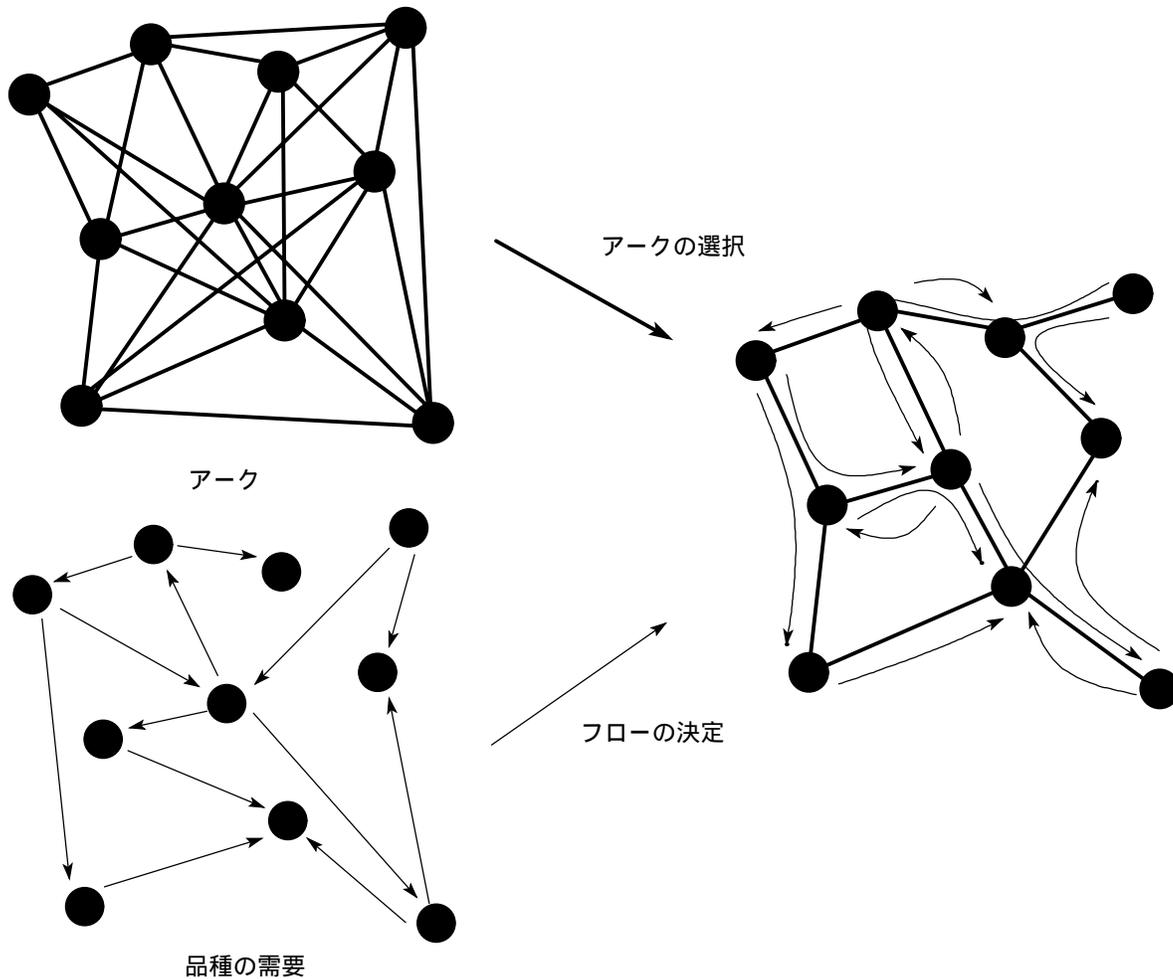


図 1.1 ネットワーク設計問題

ネットワーク設計問題における主な条件や前提を示す。

- 目的関数：費用最小化，フロー量最大化，利益最大化
- アークのデザイン費用：考慮しない，定数として考慮する，離散値として考慮する．
- アーク上のフロー費用：考慮しない，線形関数，区分的線形関数，凹関数，凸関数
- アークの向き：向きをもたない，向きをもつ
- アーク容量：考慮しない，考慮する，複数種類を考慮する．
- 品種数：単一，複数
- ノードのデザイン費用：考慮しない，考慮する．
- ノードの容量：考慮しない，考慮する．
- ネットワークの形状：木状，多層構造，リング状，一般形
- その他：多期間，輸送機器のフロー保存，ノードの経由数，結束条件，時間制約，フロー量の下限制約，サービスレベル，同一終点フローの木状制約

一般的に，デザイン費用とフロー費用を同時に考慮する問題をネットワーク設計問題と

表 1.1 ネットワーク設計問題の分類

		デザイン費用	フロー費用	フロー費用 + デザイン費用
単 一 品 種	容量制約 のない	最小木	最短路 凹費用フロー	固定費用をもつフロー
	容量制約 をもつ	容量制約をもつ最小木	最小費用フロー	容量制約をもつフロー 区分的線形費用をもつフロー
多 品 種	容量制約 のない		全対間最短路 利用者均衡フロー ハブ設計	容量制約のない設計 予算制約をもつ設計 利用者均衡条件をもつ設計
	容量制約 をもつ	ネットワーク積込	多品種フロー	容量制約をもつ設計 区分的線形費用をもつ設計

よぶ。本論文では、デザイン費用とフロー費用を同時に考慮し、ネットワークの形状が一般形であり、かつ多品種を対象とする問題を取り扱う。また、ノードの容量やデザイン費用は考慮しないが、ノードをアークに変換することによって、これらを考慮することができる。

デザイン費用またはフロー費用のみである最小木問題や最短路問題などのネットワーク問題、フロー費用のみである多品種フロー問題、凹費用フロー問題や交通ネットワーク問題などのネットワークフロー問題も、広い意味でのネットワーク設計問題の一分類と考えることもできる。表 1.1に、主なネットワーク設計問題の分類を示す。

容量制約のないネットワーク設計問題

ネットワークを設計する上で考慮すべき最も基本的な費用は、ネットワークにアークを選択するときには固定的に発生する費用であるデザイン費用と、アーク上をモノが移動するときにはフロー量に比例して発生するフロー費用である。これらの二つの費用の和を目的関数として最小化し、アーク容量を考慮しない問題を容量制約のないネットワーク設計問題 (uncapacitated network design problem; *UND*) とよぶ。アークを選択するときには固定的にかかるデザイン費用を考慮するため、この問題は固定費用をもつネットワーク設計問題ともよばれる。考慮する条件としては、各品種の需要が始点から終点に移動する、アークが選択された場合にこのアーク上をフローが移動できる、デザイン変数が 0-1 であるなどが挙げられる。

一般的に、モノが移動するための費用はそのフロー量に従って増加するが、その単価は低減することが多い。このような場合、費用は単調増加な凹関数として表現される。*UND* では、このような複雑な費用を図 1.2に示すような固定的なデザイン費用と変動的な線形であるフロー費用の和で近似していると見てもできる。

UND は、0-1 変数であるデザイン変数と連続変数であるフロー変数を含む組合せ最適化問題として定式化することができる。このため、ネットワークが完全グラフの場合、ノード数が 10~20 程度の小規模な問題であれば、CPLEX といった汎用の数理最適化ソ

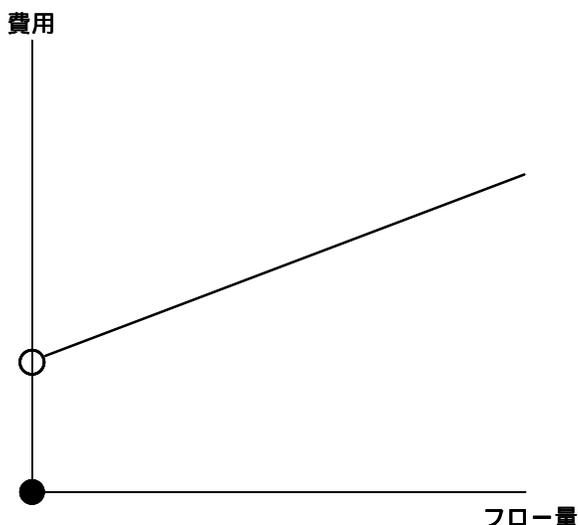


図 1.2 *UND* におけるフロー量と費用の関係

フトウエアの分枝限定法などを用いて最適解が得られる可能性がある。しかし、ノード数が 30 程度であっても、0-1 変数であるデザイン変数は 400 を超え、フロー変数は 30 万を超えてしまうことがある。この場合、一般的な数理最適化ソフトウェアでは、最適に解くことが困難となる。したがって、最適解を求める代わりに、良い実行可能解を求める近似法の開発が重要である。

UND に対する近似解法の開発の難しさの一つは、デザイン変数が確定したとしても、適切なフローを決定し、近似解の目的関数値を評価するためには、多品種の始点・終点間の最短路を求める必要があり、大きな計算量が必要となることである。一般的な近似解法では、近傍探索を数多く繰り返し、そのたびに目的関数値を評価する必要がある。*UND* では、近傍探索であるアークの削除や付加を繰り返すたびに最短路問題を解いてフロー変数を決定し、目的関数値を評価する必要があるため、全体として膨大な計算量となる。アークを削除、付加したネットワーク上の最短路問題は、Murchland 法 [63, 73] を用いることによって計算時間を短縮することができるが、それでも近傍探索に大きな計算量が必要となる。そのため、*UND* の近似解法としては、計算量を抑えることができる貪欲法が中心となる。

容量制約をもつネットワーク設計問題

容量制約をもつネットワーク設計問題 (capacitated network design problem ; *CND*) は、アーク上のフロー量がアーク容量以下であるという容量制約をもつ問題である。アーク容量は輸送能力に相当しており、*UND* に比べてより現実的なモデルとなる。考慮する費用はデザイン費用とフロー費用であり、これらの和を目的関数として最小化する。考慮する条件は、各品種の需要が始点から終点に移動する、アークが選択された場合にアーク上ではアーク容量までフローが移動できる、デザイン変数が 0-1 であるなどが挙げら

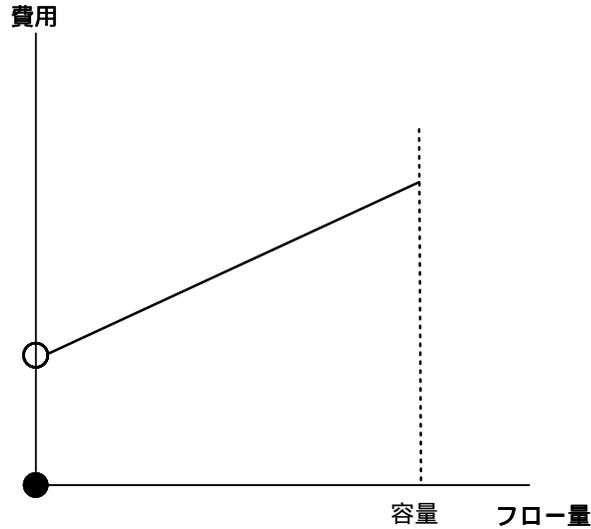


図 1.3 CND におけるフロー量と費用の関係

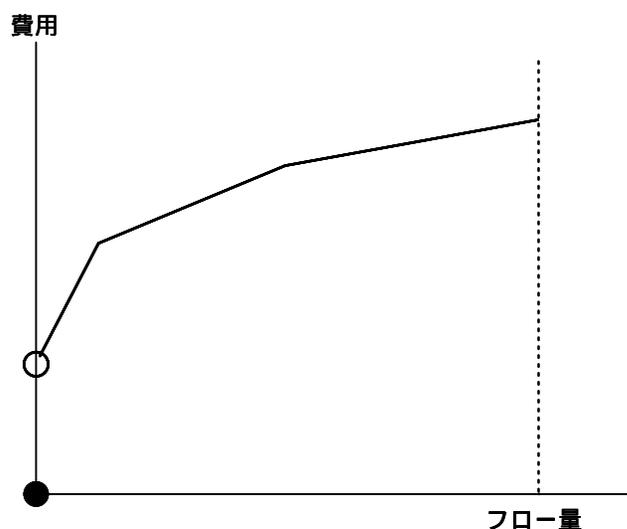
れる。

CND におけるアーク上のフロー量と費用の関係を図 1.3 に示す。デザイン変数の線形緩和を行った場合、容量制約があるために CND における費用関数と線形緩和問題における費用関数との差であるギャップが大きくなり、 UND に比べて最適値と下界値との差が大きくなる傾向がある。

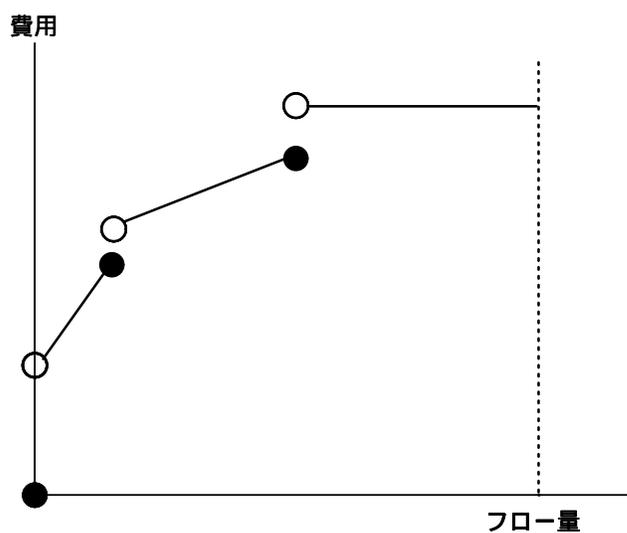
0-1 変数であるデザイン変数を決定した後にフローを求める問題は、 UND では全対間最短経路問題となるが、 CND では多品種フロー問題となり、 CND は UND よりもさらに実行可能解を求めること自体に手間がかかる。また、容量制約があるために、アークを削除していく貪欲法やアーク交換による局所探索法では、実行可能性を確保しにくい性質がある。このため、近似解法では、 UND のようにデザイン変数のみを決定変数として考えるのではなく、デザイン変数とフロー変数を同時に考慮することが必要となる。このように近似解法の面から見ても、 CND は UND よりも難しい問題となる。このようなことから、 UND に比べて、 CND では相対的に小さな規模の問題が解析の対象となる。

区分的線形費用をもつネットワーク設計問題

現実的な問題では、宅配便、混載便、貸切便などといった複数種類の輸送手段を選択することができる場合がある。このような場合には、輸送量によって費用が最小となる輸送手段を利用することになる。そのため、複数種類の輸送手段を考慮した場合、費用は処理量に対して区分的線形をもつ関数として表現できる。このような費用関数を目的関数とする問題は区分的線形費用をもつネットワーク設計問題 (piecewise linear costs network design problem; PND) とよばれる。 PND におけるフロー量と費用の関係は図 1.4 のようになる。費用関数は、必ずしも図 1.4(a) のように連続である必要はなく、図 1.4(b) のように下半連続であれば良い。考慮する条件は、各品種の需要が始点から終点に移動す



(a)



(b)

図 1.4 PND におけるフロー量と費用の関係

る，アーク上のフロー量に対応した区分のフロー費用が発生するなどが挙げられる．

区分的線形費用を目的関数とする最適化問題では， PND の費用関数と線形緩和問題における費用関数とのギャップが CND よりもさらに大きくなり，単純な方法では緩和解や下界値が非常に悪くなる傾向がある．このため， PND は CND よりもさらに難しい問題となる．

1.3.3 ロジスティクスにおける応用

ロジスティクスネットワーク設計問題は，輸送やロジスティクスなどの領域におけるロジスティクスネットワークを設計する問題を抽象化したものである．本論文で扱う問題

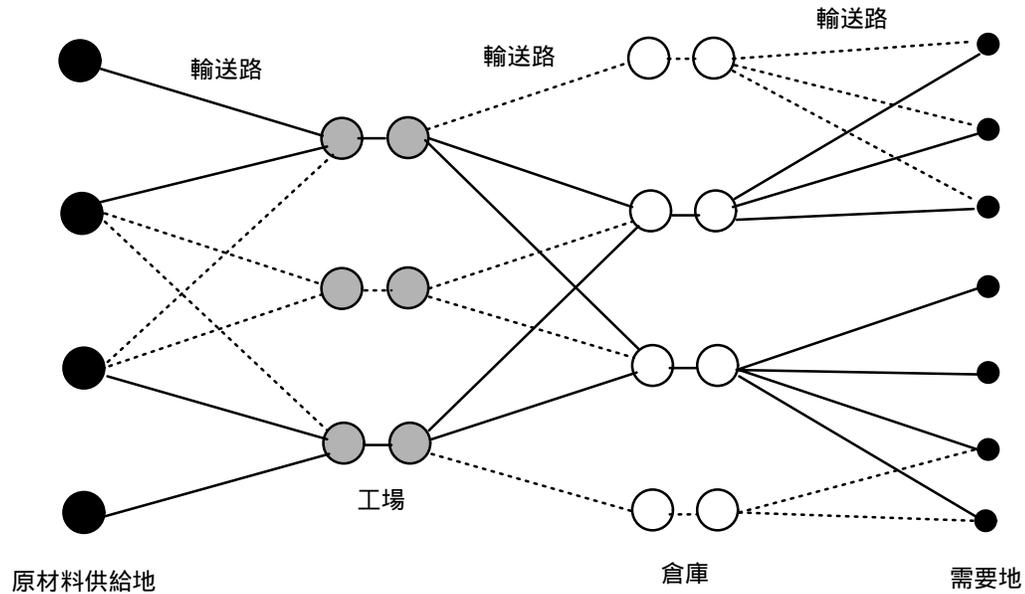


図 1.5 ストラテジックレベルのロジスティクスネットワーク設計問題

は、現実の問題を抽象化し、最も重要で基本的な条件を考慮したものである。このため、現実の問題に適用するにはさらに多くの条件を付加する必要がある。しかし、多くの条件を考慮すれば問題自体が解けず解決案を提示できないこともあり、また問題の本質からの乖離が生じる可能性もある。

ここでは、本論文で扱う問題と関連するロジスティクスにおける応用問題を記述する。

ストラテジックレベルのロジスティクスネットワーク設計問題

モノを生産し、消費者まで供給する企業では、原材料の調達、生産、輸送、保管、在庫などのロジスティクス活動を行っている。このような企業では、原材料の供給地の選択、工場や倉庫の選択、輸送路の選択といったロジスティクスネットワークの設計を行う必要がある。このような長期的な計画問題をストラテジックレベルのロジスティクスネットワーク設計問題とよぶ。

ロジスティクスネットワークを設計するには、工場の生産能力、倉庫の保管能力、輸送路の輸送能力、多品種のモノの需要などを考慮する必要がある。図 1.5 にストラテジックレベルのロジスティクスネットワーク設計問題の概念図を示す。ノードは原材料供給地、工場、倉庫、需要地であり、アークは輸送路である。工場や倉庫の選択を考慮する場合、ノードを二分してその間にアークを設定することによって、施設の選択をアークの選択に対応させることができる。

アークのデザイン費用は工場や倉庫の固定費用、輸送路の開設費用に対応し、アークのフロー費用は工場・倉庫の変動費用や輸送路の輸送費用に対応する。これらの費用の合計が最小となるネットワークを設計する。なお、アーク容量は、工場の生産能力、倉庫保管力や輸送路の輸送能力に対応する。

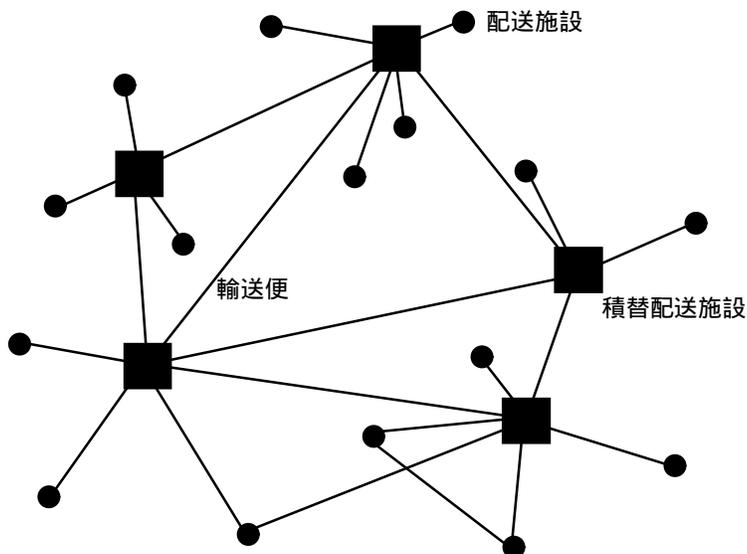


図 1.6 タクティカルレベルのロジスティクスネットワーク設計問題

タクティカルレベルのロジスティクスネットワーク設計問題

郵便や宅配便などでは、多くの地点から収集したモノを多くの地点に配達する必要がある。このため、各地域に主要配送施設を設置し、これらの中に幹線輸送便を設定して輸送を行っている。これらの企業では、どの施設間に直送便や混載便を設定し、どの施設で積替えを行うかといった輸送計画を行うことが必要である。このような中期的な計画問題をタクティカルレベルのロジスティクスネットワーク設計問題または、幹線輸送ネットワーク設計問題、混載輸送ネットワーク設計問題や Less-Than-TruckLoad 問題とよぶ。

ネットワークを設計する際には、輸送便の輸送能力、路線便の開設・運行費用や多品種のモノの需要などを考慮する必要がある。図 1.6 にタクティカルレベルのロジスティクスネットワーク設計問題の概念図を示す。ノードは配送施設であり、アークは輸送便である。アークのデザイン費用は路線便の開設費用や維持費用、アークのフロー費用は路線便における輸送費用、アーク容量は路線便の輸送能力や便数に対応する。

オペレーショナルレベルのロジスティクスネットワーク設計問題

工場や倉庫などでは、製品の需要に応じて、どの製品をいつ生産・発注し、どれだけの量を在庫するべきかという問題が発生する。このような短期的な計画問題は、生産ロットサイズ問題や調達ロットサイズ問題といったオペレーショナルレベルのロジスティクスネットワーク設計問題として表現できる。ロットサイズ問題は物理的なネットワークではないが、ネットワークモデルとして表現することができる。図 1.7 にこの問題のネットワーク表現を示す。

アークのデザイン費用は生産や発注のための固定費用、アークのフロー費用は生産や発

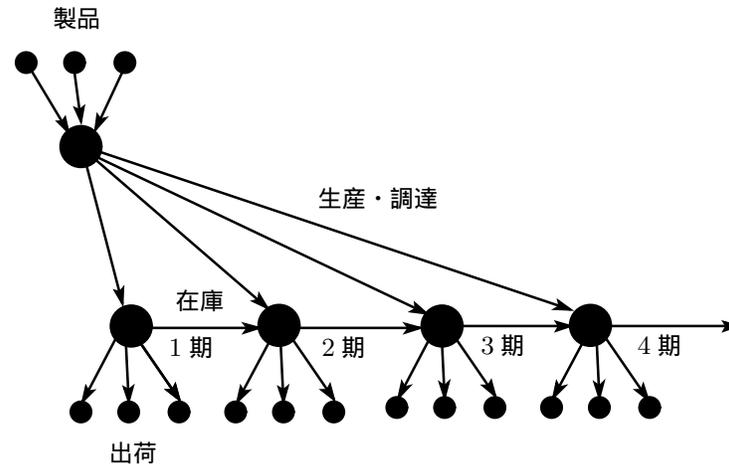


図 1.7 オペレーショナルレベルのロジスティクスネットワーク設計問題

注のための変動費用や在庫費用，アーク容量は生産，調達，在庫保管能力に対応する．

1.4 論文の構成

本論文は7章から構成されている．

第1章では，研究の背景，研究の目的，問題の提示および論文の構成を示す．問題の提示では，用語の定義およびネットワーク設計問題の定義を示し，いくつかのロジスティクスにおける応用を示す．

第2章では，従来の研究，および従来の研究の分類における本論文の各章の位置づけを示す．従来の研究では，関連する問題を容量制約のないネットワーク設計問題，容量制約をもつネットワーク設計問題，区分的線形費用をもつネットワーク設計問題などに分類し，分類毎に解説する．

第3章では，容量制約のないネットワーク設計問題を取り扱う．容量制約のないネットワーク設計問題は，アーク容量を考慮しない条件のもとで，アークを選択するときが発生する費用であるデザイン費用とモノがアーク上を移動するときが発生するフロー費用の合計を最小にするような基本的な多品種のネットワーク設計問題である．この章では，容量制約のないネットワーク設計問題に対して，アークフローを用いた定式化とパスフローを用いた定式化，強い妥当不等式，および二種類の近似解法を提案する．強い妥当不等式は，同一ノードを始点とする品種と終点とする品種のフローが，同一のアーク上で同一方向に移動するフローが存在しないことを用いた非順流不等式である．提案する第一の解法は，不要であろうアークを順次除いていく貪欲法である．この解法は，従来の貪欲法を厳密化した方法であり，大規模な問題を高速に解くことができる解法である．第二の解法は，強い妥当不等式を含む定式化の線形緩和にもとづく線形緩和法である．これは，パスフロー変数を用いた定式化とパスフローに関する列生成法，行生成法と分枝限定法を組み合わせた解法であり，より高精度の解を求めることができる．最後に，数値実験を行い，

従来の解法の結果と比較することによって、提案した定式化と解法の有効性を検討する。

第4章では、第3章と同じ容量制約のないネットワーク設計問題を取り扱う。この章では、容量制約のないネットワーク設計問題に対する三種類の Lagrange 緩和法を示す。Lagrange 緩和法は、Lagrange 緩和問題とその解法、Lagrange 乗数の更新、Lagrange ヒューリスティックを組み合わせた手法であり、組合せ最適化問題に適用されて成果を挙げている解法である。第一の緩和法は、従来の Lagrange 緩和法を向きをもたない問題に適用した方法である。Lagrange ヒューリスティックでは、第3章で提案した貪欲法を用いる。第二の緩和法は、始点を同じにする品種に対する非逆流不等式を制約式に含めた強い定式化に対する Lagrange 緩和法である。第三の緩和法は、非逆流不等式と非順流不等式を含むより強い定式化に対する Lagrange 緩和法である。最後に、数値実験を行い、従来の解法の結果と比較することによって、提案した三種類の Lagrange 緩和法の有効性を検討する。

第5章では、容量制約をもつネットワーク設計問題を取り扱う。容量制約をもつネットワーク設計問題は、アーク上のフロー量がアーク容量以下であるという容量制約をもつ問題である。アーク容量を考慮しなければならないため、この問題は、容量制約のないネットワーク設計問題よりも難しい問題となる。この章では、アークフローを用いた定式化とパスフローを用いた定式化を示した後、容量制約をもつネットワーク設計問題に対する二種類の容量スケールリング法を用いた近似解法を示す。容量スケールリング法は、その線形緩和問題から得られる解を用いて容量を逐次変化させ、0-1 解を導く方法である。第一の解法は、アークフロー変数を用いた二種類の定式化にもとづく容量スケールリング法である。第二の解法は、パスフロー変数を用いた定式化にもとづく容量スケールリング法であり、アークフロー変数を用いた定式化よりも高速に解を求めることができる。この解法では、列生成法、行生成法と分枝限定法を組み合わせた方法を用いる。最後に、数値実験を行い、従来の解法の結果と比較することによって、提案した容量スケールリング法の有効性を検討する。

第6章では、区分的線形費用をもつネットワーク設計問題を取り扱う。区分的線形費用をもつネットワーク設計問題は、複数種類の輸送手段を選択することができるモデルに対応しており、容量制約をもつネットワーク設計問題よりもさらに難しい問題となる。この章では、区分的線形費用をもつネットワーク設計問題に対して、アークフローおよびパスフローを用いた定式化を示した後、容量スケールリング法を用いた二種類の近似解法を示す。第一の解法は、アークフロー変数を用いた三種類の定式化にもとづく容量スケールリング法と分枝限定法を組み合わせた解法である。第二の解法は、パスフロー変数を用いた定式化にもとづく容量スケールリング法であり、列生成法、行生成法と分枝限定法を組み合わせた方法であり、アークフロー変数を用いた定式化よりも高速に解を求めることができる。最後に、数値実験を行い、従来の解法の結果と比較することによって、提案した容量スケールリング法の有効性を検討する。

第7章では、第2章から第6章までに得られた成果をまとめ、本論文の結論を示す。

第 2 章

従来の研究

本章では、容量制約のないネットワーク設計問題，容量制約をもつネットワーク設計問題，区分的線形費用をもつネットワーク設計問題，およびネットワーク設計問題に関するサーベイに対する従来の研究を記述する．続いて，本論文と従来の研究との関連，および多品種を対象としたネットワーク設計問題に対する研究における本論文の研究の位置づけを示す．

2.1 従来の研究

ネットワーク設計問題に対する研究は，1960年代末に交通ネットワークの分野から始まり，1970年代からは通信ネットワーク，サプライチェーンやロジスティックネットワークの分野が加わり，今日に至っている．この節では，従来の研究の概要，個々の研究，およびサーベイの要約を示す．

2.1.1 容量制約のないネットワーク設計問題

容量制約のないネットワーク設計問題 (UND) は，デザイン費用とフロー費用の和を目的関数とした問題である．しかし，歴史的に見るとネットワーク設計問題の研究は，デザイン費用に対する予算が与えられ，この予算制約のもとでフロー費用の和の最小化する問題である予算制約をもつネットワーク設計問題 (budget network design problem; BND) の研究から始まっている． UND は， BND において予算制約を Lagrange 双対化した問題と見ることができ，二つの問題に共通的に適用可能な解法も数多く提案されている．そのため，ここでは BND に対する研究も併せて記述する．

UND および BND に対する従来の研究は，近似解法，最適解法，緩和法および解析的研究の四種類に分けることができる．

巡回セールスマン問題や配送経路問題などといった他の組合せ最適化問題に比べて， UND や BND は実行可能解の目的関数値を評価するための計算量が大きな問題である．このため，近似解法としては貪欲法であるフォワード法やバックワード法が研究の中心で

ある．Scott[71] は BND に対するフォワード法およびバックワード法を初めて提案している．その後，Dionne-Florian[26]，森津 [62] らはフォワード法またはバックワード法の改良法を示しており，Boffey-Hinxman[10] は分枝限定法に制限を加えた解法を示している． UND に対しては，Billheimer-Gray[9]，Los-Lardinois[56] はフォワード法およびバックワード法を示し，Minoux[60] は高速化したバックワード法を示している．

最適解法としては分枝限定法によるものが多数を占める． BND に対して，Scott[71] は下界値を用いないバックトラック法を示している．分枝限定法を用いた研究として Boyce-Farhi-Weischedel[11]，Hoang[41]，Dionne-Florian[26]，Boffey-Hinxman[10]，森津 [62] が解法を示している．一方， UND に対して，Los-Lardinois[56] は分枝限定法，Magnanti-Mireault-Wong[58] は Benders 分解法を示し，Guieérrez-Kouvelis-Kurawarwala[39] は Benders 分解法を確率モデルに適用している．

緩和法としては， BND に対して，Gallo[29] および Ahuja-Muty[1] は下界平面を目的関数とした緩和問題を示し，片山-春日井 [46] は Lagrange 緩和法を示している． UND に対して，Balakrishnan-Magnanti-Wong[7] は双対上昇法，Lamar-Sheffi-Powell[55] は容量改善法，Holmberg-Hellstrand[42] は Lagrange 緩和法を示している．

解析的な研究として，Johnson-Lenstra-RinnooyKan[44] は決定問題の NP 完全性の証明，Wong[74] はワーストケース解析，Wong[77] は漸近的解析，Balakrishnan[5] は多面体的解析を行っている．

Scott(1969)[71]

BND ：バックトラッキング法を用いた最適解法と二つの近似解法を提案している．バックトラッキング法は下界値を用いない深さ優先の探索法である．近似解法は貪欲法であり，すべてのアークを付加または削除したときの目的関数値の変化量を求め，適当な初期解から予算制約を満足するように目的関数値の変化量が最小または最大となるアークを付加または削除していく方法である．この解法はフォワード法およびバックワード法とよばれる．ユークリッド平面上の 10 ノード，45 アークまでの問題について数値実験を行い，最適解と近似解を求めている．近似解法により，80～90%の問題で最適解を得ている．

Boyce-Farhi-Weischedel(1973)[11]

BND ：予算制約を取り除くという緩和問題を提案し，未決定のすべてのアークを選択したときの総フロー費用を下界値としている．また，この下界値を用いた分枝限定法による最適解法を提案している．10 ノード，45 アークについて数値実験を行い，大半の問題において最適解を求めている．

Hoang(1973)[41]

BND ：アークを取り除いたときのこのアークの両端点を始点・終点とする品種のフロー費用の増加分を目的関数の係数とし，予算制約式のみを制約式としたナップサック型の緩和問題を提案している．また，この緩和法を用い，分枝変数を小数変数とした分枝限定法による最適解法を提案している．9 ノード，20 アークまでの問

題について数値実験を行い、最適解を求めている。

Billheimer-Gray(1973)[9]

UND : フォワード法およびバックワード法を提案している。このバックワード法は、はじめに各アークの両端間について、最短距離と第二最短路の距離より当該アークを削除した場合の目的関数値の変化量を近似的に計算しておき、この値が最大値のアークを削除していく方法である。フォワード法は、はじめにアークを付加した場合の目的関数値の変化量を近似的に計算しておき、この値が最大値のアークを付加していく方法である。また、釘付けテストも提案している。8 ノード, 23 アークの数値例と 225 ノード, 900 アークの計算時間を示している。

Minoux(1976) [60]

UND : バックワード法を提案している。この解法では、アーク上のフローのパスをこのアークを取り除いたときの両端点間の最小費用であるパスに代えたときの費用の減少量を求め、この減少量の大きなアークから取り除いていく。アークが削除の対象になるときにのみ減少量を再計算することによって、計算量を大きく減少させている。8 ノード, 15 アークまでの問題の数値例を示している。

Johnson-Lenstra-RinnooyKan(1978)[44]

BND : ナップサック問題の決定問題に帰着できることを用いて、*BND* の決定問題が *NP* 完全であることを示している。さらに、すべてのデザイン費用とアーク上のフロー費用が 1 で、予算が「ノード数 -1」である特殊な決定問題も *NP* 完全であることを示している。

Dionne-Florian(1979)[26]

BND : Hoang 法における最短路探索と分枝変数の選択法を改良した最適解法と近似解法を提案している。最短路問題の解法には、Murchland 法を用いている。また、単位デザイン費用当たりの目的関数の変化値が最大である変数を分枝変数としている。さらに、単位デザイン費用当たりの目的関数の変化値をアークの削除基準としたバックワード法と近似下界値を用いた分枝限定法による近似解法を提案している。29 ノード, 54 アークまでの問題について数値実験を行い、最適解法では Boyce 法よりも短時間で最適解を得ることができ、近似解法では大半の問題に対して最適解を得ている。

Boffey-Hinxman(1979)[10]

BND : 近似解法と分枝限定法による最適解法を提案している。近似解法は分枝ノード数に制限を加えた分枝限定法である。最適解法では、単位デザイン費用当たりの目的関数の変化値が最大となる変数を分枝変数とする方法と最小となる変数を分枝変数としている。また、釘付けテストも提案している。20 ノード, 190 アークまでの問題について数値実験を行い、近似解法では大半の問題で最適解を得ている。また、最適解法により、予算が大きな 20 ノードの問題を解いている。

Wong(1980)[74]

BND : ワーストケースにおける近似解法の解析を行っている。ワーストケースの

精度がノード数の 1 次関数未満である決定問題が NP 完全であることを示している．最小木を用いた近似解法のワーストケースにおける精度がノード数の 2 倍であることを示している．また，デザイン費用が 1 である問題では，フロー費用をアークの長さとしたネットワークにおける最短路木を近似解とする解法のワーストケースの精度が 2 であることを示している．

Los-Lardinois(1982)[56]

UND：分枝限定法を用いた最適解法とフォワード法およびバックワード法を提案している．分枝限定法では，アークを削除したときの両端点間のデザイン費用の減少量と両端点を始点・終点とする品種のフロー費用の増加量の値の和から算出した下界値を用いる．近似解法は，Billheimer 法を発展させた解法である．バックワード法では，最初にアークを削除した場合の目的関数値の変化量を厳密に求め，単位デザイン費用当たりのフロー費用を基準にアークを削除している．12 ノード，40 アークの問題を最適に，30 ノード，93 アークの問題を近似的に解いている．また，15 ノード，105 アークの問題について数値実験を行い，統計的に解析している．

Gallo(1983)[29]

BND：従来の緩和問題を下界平面の概念によって整理し，品種分割方法とアーク分割方法によって異なる下界平面が定義できることを示している．さらに，アークを取り除いたときの未選択品種間のフロー費用の増加分を目的関数の係数とした二種類の下界平面の算出法を提案し，Hoang の下界値よりも提案した下界値が良いことを示している．29 ノード，54 アークまでの問題について数値実験を行い，Hoang の下界平面による下界値との比較を行い，Hoang の下界値に比べて 3 % 程度良い下界値を得ている．

森津 (1984)[62]

BND：Scott 法に分枝子問題の実行可能性を考慮した終端操作の改良を加えた最適解法を提案している．また，目的関数値の変化量を近似的に計算した簡易フォワード法とバックワード法，同時に数本のアークを選択する段階的フォワード法とバックワード法，および局所最適解探索後に近傍最適化を行う近似解法を提案している．9 ノード，23 アークまでの問題に対して最適解を求め，81 ノード，261 アークまでの問題に対して近似解を求めている．

Wong(1985)[77]

BND：近似解法は確率的，漸近的な解析を行っている．平面上の中心ノードを中心とするスター型ネットワークまたは最小木に一定値以下の長さのアークを加えるという近似解法を提案し，この解法によって得られるフロー費用がノード数の増加とともに，最適値に漸近していくことを示している．ノード数 150 までの問題について数値実験を行い，0 ~ 35 % の精度の解が得られることを示している．

Magnanti-Mireault-Wong(1986)[58]

UND：Boyce，Hoang，Gallo の下界平面を Benders カットの視点から整理し，従来の Benders カットよりも強い Benders カットとパレート最適カットを提案し

ている．また，問題の規模を縮小するための前処理方法も示している．提案した Benders カットと Benders 分解法を用いて，30 ノード，90 アークまでの問題について数値実験を行い，大半の問題の最適解を得ている．

Balakrishnan(1987)[5]

UND：パスフローを用いて定式化した問題の線形緩和問題の性質を検討している．線形緩和問題において，フロー変数の最大値がデザイン変数に一致し，品種対補グラフ行列が一次独立の場合に限り，解が端点解になることを示している．さらに，この性質を用いてフロー変数とデザイン変数に関する二種類の切除平面を提案している．

Ahuja-Muty(1987)[1]

BND：Gallo の下界平面の係数はアークの選択順に大きく依存するため，アーク選択順に依存されにくい下界平面の算出法を提案し，この下界平面による下界値が Gallo による下界値よりも良いことを示している．また，計算量の小さい下界平面の算出法も提案している．50 ノード，100 アークまでの問題について数値実験を行い，Hoang および Gallo の下界値との比較を行い，1~3 %程度良い下界値を得ている．

Balakrishnan-Magnanti-Wong(1989)[7]

UND：始点または終点を同じにする品種がアーク上を逆流しないことを利用した強い制約式を提案している．また，連続緩和問題の双対問題を近似的に解くための双対上昇法を提案している．この双対上昇法は Dijkstra 法を *UND* に拡張したラベリング型の解法である．また，問題を縮小するための釘付けテストを示し，双対解を初期解として，フォワード法およびバックワード法で改善する双対ヒューリスティックを提案している．提案した双対上昇法と近似解法を用いて，35 ノード，595 アークまでの問題について数値実験を行い，誤差 0~3 %の上界値と下界値を求めている．

Lamar-Sheffi-Powell(1990)[55]

UND：有向アークを対象としている．アーク上の最適フロー量と擬似的なアーク容量の関係に注目し，最適フロー解を満足する範囲内でアーク容量を減少させ，下界値を上昇させる容量改善法を提案している．線形緩和問題が線形フロー問題に帰着されることを示し，適当な目的関数の目標値とアーク容量の試行値を用いて妥当なアーク容量制約を求め，妥当なアーク容量制約を追加することによって下界値を改善している．

片山-春日井(1995)[46]

BND：フロー保存式を Lagrange 緩和した緩和問題がナップサック問題となることを示し，劣勾配法を用いた下界値の解法と緩和問題を用いた Lagrange ヒューリスティックを示している．75 ノード，2775 アークまでの問題について数値実験を行い，誤差 0~3 %の上界値と下界値を求めている．Gallo 法，バックワード法との比較も行っている．

Guiérrez–Kouvelis–Kurawarwala(1996)[39]

UND : フロー費用が確率的に変化する問題を取り扱い, 解法の頑強性について検討している. シナリオにもとづく複数の入力に対して, 頑強性問題に Benders 分解法を適用する二つの近似解法を示している. 33 ノード, 100 アーク, 64 品種の問題について数値実験を行っている.

Holmberg–Hellstrand(1998)[42]

UND : 向きをもたない問題を対象としている. フロー保存条件に対する Lagrange 緩和法, および Lagrange 緩和問題による下界値を用いた分枝限定法を示している. また, 緩和問題のデザイン変数解を用いて, 多品種フロー問題を解く Lagrange ヒューリスティックを示している. 150 ノード, 600 品種までの品種の問題について数値実験を行っている.

2.1.2 容量制約をもつネットワーク設計問題

容量制約をもつネットワーク設計問題 (*CND*) に対する従来の研究は, 大きく多面体解析, 緩和法, 近似解法の三種類に分けることができる.

多面体解析として, 問題の実行可能領域を満足し, 緩和問題の実行可能領域の一部を満足しない制約式である妥当不等式の提示がある. 片山–春日井 [45] はカットセット上のフローに関する制約からカットセット不等式を示している. Magnanti–Mirchandani–Vachani[57] はカットセット不等式, 三分割不等式および剰余容量不等式を示している. Barahona[8] は多カットセット不等式を示している. また, Chouman–Crainic–Gendron[14] はカットセット不等式, 被覆不等式, 最小基数不等式, およびこれらの持ち上げによる妥当不等式を示している.

緩和法としては, 線形緩和と Lagrange 緩和が用いられている. しかし, *UND* と比べて, 定式化自体が弱く, 緩和問題から得られる下界値は相対的に良くない. 片山–春日井 [45] はカットセット不等式を用いた定式化に対する双対上昇法を示し, Gendron–Crainic[31, 33] および Gendron–Crainic–Frangioni[32] は線形緩和問題と Lagrange 緩和問題, およびその解法を示し, Crainic–Frangioni–Gendron[17] は束法を用いた劣勾配法を提案している. また, Holmberg–Yuan [43] は Lagrange 緩和と分枝限定法を組合せた解法を示し, Herrmann–Ioannou–Minis–Proth[40] は *UND* に対する双対上昇法を拡張した解法を提案している.

近似解法としては, 初期には貪欲法, 2000 年以降はタブー探索法をはじめとするメタ解法が用いられている. Gendron–Crainic [32, 33] は多品種フロー問題に対する資源主導にもとづく解法を応用した近似解法を示している. Crainic–Gendreau–Farvolden[18] および Zaleta–Socarrás[78] は単体法にもとづいたタブー探索法を提案し, Ghamlouche–Crainic–Gendreau [36] はサイクルにもとづいたタブー探索法を提案している. Ghamlouche–Crainic–Gendreau [37] および Álvarez–González–De-Alba [2, 3]

はパス再結合法や散布法とよばれるメタ解法を提案している．Crainic–Gendreau [19] は共同タブー探索法，Crainic–Li–Toulouse [21] は多レベル共同タブー探索法を提案している．Crainic–Gendron–Hernu [20] は傾斜スケールリング法を提案し，Pochet–Vyve[66] は単一品種問題に容量スケールリング法を適用している．

片山直登，春日井博 (1993)[45]

カットセット上のフローに関する制約の丸めによるカットセット不等式を示し，このカットセット不等式を用いた定式化に対する双対上昇法を示している．双対上昇法では，線形緩和問題の双対問題を作成し，各品種のパスが最短パスとなる条件を満足する範囲内で双対変数を効率的に設定し，下界値を上昇させている．また，双対上昇法との比較にフロー保存式に対する Lagrange 緩和法を用いている．14 ノード，91 アークまでの問題について数値実験を行っている．

Magnanti–Mirchandani–Vachani(1993) [57]

多面体解析を行っている．3 ノードの問題に対して，カットセット不等式から三分割不等式を導いている．また，フロー保存式を Lagrange 緩和した単一アーク問題に対するファセットおよびアーク剰余容量不等式を示している．

Gendron–Crainic–Frangioni(1994)[32]

品種毎に非集約化した強制制約式を含む定式化について，フロー保存式に対する Lagrange 緩和法，容量制約式と強制制約式に対する Lagrange 緩和問題を示し，劣勾配法を用いた解法を示している．100 ノード，1600 アーク，200 品種までの問題について数値実験を行っている．

Gendron–Crainic(1994)[31]

強制制約式を含む定式化について，強制制約式に対する Lagrange 緩和問題とその解法，およびデザイン変数の緩和解を用いた資源分解ヒューリスティックを示している．さらに，並列計算機の環境における上界値と下界値の並列計算手法を示している．二部グラフ上の 100 ノード，2500 アーク，100 品種までの問題について数値実験を行っている．

Barahona(1996) [8]

フロー変数が整数である問題を対象としている．カットセット上のフローに関する制約の丸めによるカットセット不等式を示している．さらに，ネットワークを分割したノード集合間に跨るアーク数の制約である多カット不等式およびその生成法を示している．64 ノード，2016 品種までの問題について数値実験を行っている．

Herrmann–Ioannou–Minis–Proth(1996) [40]

Balakrishnan–Magnanti–Wong が示した *UND* に対する双対上昇法を容量制約をもつ問題に拡張している．双対上昇法の過程において，スラック変数が 0 となるデザイン変数により構成されるネットワークを求め，このネットワーク上で多品種フロー問題を解き，容量が不足するアークがあればフロー費用にペナルティーを加えて，双対上昇法を解き直す方法である．

Gendron-Crainic(1996)[33]

Lagrange 緩和解をもとにした資源分解ヒューリスティックを示している。これは、射影問題を解くことによってデザイン変数を定め、緩和された容量制約式を満足しかつフロー解の近傍にある解を求め、フローが正であるアークに対して一定の基準を用いてアークを削除していく方法である。100 ノード、700 アーク、400 品種までの問題について数値実験を行い、資源分解ヒューリスティックと CPLEX による分枝限定法と比較している。

Gendron(1999) [30]

Herrmann らが示した双対上昇法により算出した値が下界値とならない例を示し、彼らの示した下界値の解法が妥当でないことを指摘している。

Crainic-Gendreau-Farvolden(2000) [18]

パスフローを用いた定式化を使用し、単体法における基底解であるパスフローにもとづいたタブー探索法を示している。タブー探索法では、連続近傍に対する局所探索に短期メモリを、離散近傍に対する多様化に長期メモリを適用している。100 ノード、700 アーク、400 品種までの問題について数値実験を行い、貪欲法および資源分解ヒューリスティックと比較している。

Crainic-Frangioni-Gendron(2001) [17]

二種類の Lagrange 緩和問題を示し、劣勾配を求める手法として束法を用いた解法を示している。束法は、過去の劣勾配と現在の劣勾配から、二次計画問題を解くことによってノルムが最小となる降下方向を求める手法である。100 ノード、700 アーク、400 品種までの問題について数値実験を行い、通常の劣勾配法および CPLEX の分枝限定法と比較している。

Holmberg-Yuan(2000)[43]

フロー保存条件に対する Lagrange 緩和問題とその解法、および分枝限定法を示している。Lagrange 緩和問題では、カットセット不等式をカットとして追加している。分枝限定法における下界値の算出には Lagrange 緩和問題を用いており、厳密法と変数の近似的固定法を用いた近似解法を示している。150 ノード、1000 アーク、282 品種までの問題について数値実験を行っている。

Crainic-Gendreau(2002) [19]

単体法における基底解であるパスフローに対するタブー探索法を並列計算モデルに拡張した共同解法を提案している。この解法では、適当な解集合をプールしておき、この集合から六種類の基準により初期解を取り出し、初期解をタブー探索法により改善し、得られた局所解をプールするという手順を繰り返す。100 ノード、700 アーク、400 品種までの問題について数値実験を行い、六種類の基準と共同解法でないタブー探索法を比較している。

Ghamlouche-Crainic-Gendreau(2003) [36]

サイクルにもとづいた近傍を用いたタブー探索法を示している。これは、ネットワーク上の負の費用をもつサイクルを見つけ、このサイクル上でフローを変更し、

複数のアークの付加や削除を同時に行うという近傍探索にもとづいた方法である．100 ノード，700 アーク，400 品種までの問題について数値実験を行い，パスフローにもとづいたタブー探索法と比較している．

Chouman–Crainic–Gendron(2003) [14]

いくつかの妥当不等式とこれらを持ち上げた不等式を示している．カットセットに対する最小被覆から被覆不等式を導き，カットセット上に存在すべき最小のアーク数から最小基数不等式を導いている．さらに，ネットワークカットセット不等式も示している．これらの不等式を用いた切除平面法を示し，100 ノード，700 アーク，400 品種までの問題について数値実験を行い，線形緩和問題による下界値と比較している．

Crainic–Gendron–Hernu(2004) [20]

線形緩和フロー解をもとにフロー費用を変化させて緩和問題を解き直すという傾斜スケールリング法を示している．また，フロー保存式，容量制約式および強制制約式を緩和した Lagrange 摂動を用いた傾斜スケールリング法，および長期メモリを用いた傾斜スケールリング法も示している．100 ノード，700 アーク，400 品種までの問題について数値実験を行い，CPLEX，パスフローにもとづいたタブー探索法，サイクルにもとづいたタブー探索法，パス再結合法と比較している．

Ghamlouche–Crainic–Gendreau(2004) [37](2004)

参照解集合から初期解とガイド解を選び，これらの解から新たな解を生成していくパス再結合法を提案している．参照解集合は，サイクルにもとづいたタブー探索法によって求めた解集合から選択している．また，六種類の参照解集合の選択方法と，六種類の初期解とガイド解の選択法を示している．700 アーク，400 品種まで問題について数値実験を行い，数値実験を行っている．

Pochet–Vyve(2004) [66]

単一品種問題：線形緩和フロー解にもとづいてアーク容量を変更した線形緩和問題を解き直し，デザイン変数が 0 または 1 になるまで繰り返すという容量スケールリング法を提案している．ロットサイズ決定問題，1 始点・1 終点のネットワーク設計問題，多始点・多終点をもつ単一品種のネットワーク設計問題について数値実験を行い，制限時間つきの分枝限定法などの解法と比較している．

Zaleta–Socarrás(2004) [78]

向きをもたないアークのモデルに対してタブー探索法を適用している．タブー探索法は Crainic–Gendreau–Farvolden とほぼ同一である．50 ノード，100 品種までの問題について数値実験を行っている．

Álvarez–González–De-Alba(2005) [3]

散布探索法を用いた解法を示している．散布探索法はパス再結合法ともよばれ，参照解集合の解の組合せによって新たな解を生成し，改善していく方法である．解の改善には，剰余デザイン費用とフロー費用によるパス費用を求め，これにもとづきフローを入れ替える方法を用いている．50 ノード，100 品種までの問題について

数値実験を行い，CPLEX および双対上昇法と比較している．

Álvarez-González-De-Alba(2005) [2]

ランダム化貪欲適応探索法を散布探索法に組み込んだ解法を示している．ランダム化貪欲適応探索法は，多スタート局所探索法における局所探索法の初期解の生成過程にランダム性を加味した貪欲法を用いる方法である．50 ノード，100 品種までの問題について数値実験を行い，CPLEX および双対上昇法と比較している．

Crainic-Li-Toulouse(2006) [21]

並列計算モデルである共同解法を多レベルに拡張した解法を示している．この解法では，固定したアーク数をレベルと定義し，各レベルのエリート解を初期解とした隣接するレベルにおけるサイクルにもとづいたタブー探索法を行い，解を改善する方法である．700 アーク，400 品種までの問題について数値実験を行い，サイクルにもとづいたタブー探索法およびパス再結合法と比較している．

2.1.3 区分的線形費用をもつネットワーク設計問題

区分的線形費用をもつネットワーク設計問題 (*PND*) の研究は，近年，始められたばかりであるため，従来の研究は比較的少ない．単一品種の区分的線形費用をもつ問題に対して，Kim-Pardalos[50, 51] は傾斜スケールリング法，Kim-Pan-Pardalos[49] は傾斜スケールリング法とタブー探索法を組み合わせた解法を示し，Croxtton-Gendron-Magnanti[22] は三種類の定式化とそれらの性質を分析している．多品種の問題に対して，Croxtton-Gendron-Magnanti[23] は物流センターにおける積み替えモデルに適用し，Muriel-Munshi[64] は線形緩和にもとづく解法を示し，Croxtton-Gendron-Magnanti[24] は三種類の定式化の幾何学的な解釈と近似解法を示している．

Kim-Pardalos(2000) [51]

単一品種問題：区分的線形費用をもつ問題に対して，フロー費用を変化させた線形ネットワークフロー問題を繰り返し解く傾斜スケールリング法を提案している．複数の区分に対して動的領域構築法とよぶ区分決定法を示している．37 ノード，335 アーク，5 区分までの問題について数値実験を行い，CPLEX による分枝限定法と比較している．

Kim-Pardalos(2000) [50]

単一品種問題：アークを区分毎のアークに多重化することによって，区分的線形費用をもつ問題が等価な固定費用をもつネットワークフロー問題となることを示し，この問題の解と元の問題の解の関係を分析している．37 ノード，335 アーク，5 区分までの問題について数値実験を行い，CPLEX による分枝限定法と比較している．

Croxtton-Gendron-Magnanti(2003) [22]

単一品種問題：区分的線形費用をもつ問題に対して，増分モデル，複数選択モデル，

凸組合せモデルによる三種類の定式化を示している．これらの定式化の線形緩和問題が等価な問題となることを示し，Lagrange 双対問題における最適解の性質を分析している．

Croxton–Gendron–Magnanti(2003) [23]

三種類の定式化を用いて，多期間の物流センターにおける積み替えモデルに適用している．緩和問題に有効な妥当不等式を追加して下界値を改善する切除平面法，線形緩和問題を用いて 0–1 変数を決定する丸め込みヒューリスティック，および分枝限定法を示している．9 工場，10 物流施設，73 顧客，10 品種，6 期間，13 区分までの問題について数値実験を行い，CPLEX と比較している．

Muriel–Munshi(2004)[64]

品種と区分の両方を非集約化した強制制約式を含む拡張モデルを提案している．アークを通過する需要の比率を固定した問題に対して，解の性質と解法を示している．また，線形緩和問題において，品種のフローをマージナル費用にしたがって流し代えて解を改善する近似解法を示している．40 ノード，400 アーク，60 品種までの問題について数値実験を行っている．

Kim–Pan–Pardalos(2006) [49]

単一品種問題：傾斜スケール法とタブー探索法を組み合わせた解法を示している．短期メモリに変更したアークを格納し，長期メモリは被約費用を用いて設定している．210 ノード，10973 アークまでの問題について数値実験を行っている．

Croxton–Gendron–Magnanti(2007) [24]

区分的線形費用をもつ問題に対して，基本モデル，品種毎に非集約化した強制制約式を含む強いモデル，および品種と区分の両方を非集約化した強制制約式を含む拡張モデルを提案している．これらの定式化の幾何学的な解釈と性質の分析を行い，線形緩和問題と閾値を用いた近似解法を示している．25 ノード，150 アーク，10 品種までの問題について数値実験を行っている．

2.1.4 応用問題

ネットワーク設計問題を現実的な問題に適用させた研究が数多く存在する．ここでは，主なロジスティクスに関連する応用問題，およびそれらに対する主な研究を示す．

輸送の分野では，少量貨物の積み替えを考慮した輸送計画問題があり，積み替えターミナルの配置，輸送便や回送便の計画を同時に行なう問題は，Less-Than-Truckload 問題とよばれる．Powell[67] は輸送頻度サービス水準を条件とする問題を取り扱い，Powell–Sheffi[68] は APPOLO，Roy–Crainic[69] は NETPLAN とよばれる解析システムを開発している．

航空輸送の分野において，ハブ空港の配置とそれに接続する地方空港および航空路を決定する問題は，ハブネットワークデザイン問題とよばれる．O’Kelly[65] は第一ハブ第二ハブ割当法，Klincewicz[52] は単一交換法と複数交換法，Campbell[13] は最大フロー法

と割当フロー法を提案している。また, Skorin-Kapov-Skorin-Kapov-O'Kelly[72] はタブー探索法, Ernst-Krishnamoorthy[27] はアニーリング法, Ernst-Krishnamoorthy[28] 切除平面法を用いた解法を提案している。

工場や倉庫の立地, 多品種の生産や輸送問題を同時に考慮する問題は, サプライチェーンネットワーク設計問題とよばれる。Geoffrion-Graves[35] は Hunt-Wesson Foods 社, Brown-Graves-Honczarenko[12] は NABISCO 社に対する事例分析を行っている。

世界的規模を対象とするモデルは, グローバルサプライチェーンネットワーク設計問題とよばれ, 為替, 移転価格などを考慮している。この問題に対して, Arntzen-Brown-Harrison-Trafton[4] は DEC 社を対象とした事例分析を行い, Goetschalckx-Vidal-Dogan[38] は移転価格を含むモデル, Santoso-Ahmed-Goetschalckx-Shapiro[70] は確率計画モデルを提案している。

Powell(1986) [67]

頻度サービス水準を考慮した Less-Than-Truckload 問題を扱っている。ネットワーク上で輸送経路を付加, 削除する効率的な方法を示し, その後貨物経路を再最適化する近似解法を示している。300 施設までの問題を解いている。

Powell-Sheffi(1989) [68]

Less-Than-Truckload 問題に対する解析システムである APOLLO を開発している。このシステムは, 分解法, ヒューリスティックとユーザインターフェイスからなり, ネットワーク設計の構造を含む大規模な混合整数計画問題に対して, 近似解を提供することができる。

Roy-Crainic(1992) [69]

積み替えを考慮した Truckload および Less-Than-Truckload 問題に対するネットワークフロー最適化システムである NETPLAN を提供している。このシステムは, 輸送経路, 頻度サービス, 輸送手段, 施設における容量水準などを含む中期的な計画モデルを扱うことができる。

O'Kelly(1987) [65]

ハブネットワーク設計問題を 0-1 変数による 2 次計画問題を用いた定式化を示している。この問題に対して, ハブノード選択後に最近のハブに非ハブノードを割当てる最近ハブ割当法と一番目と二番目に近いハブに割当てる第一ハブ第二ハブ割当法を提案している。

Klincewicz(1991) [52]

単一ハブネットワーク設計問題に対して, ハブノード選択後に一つハブノードと一つ非ハブノードを交換する単一交換法を提案し, 効率的な目的関数の変化量の計算方法を示している。さらに, 二つハブノードと二つ非ハブノードを交換する複数交換法も提案している。

Skorin-Kapov-Skorin-Kapov-O'Kelly(1994) [72]

単一ハブネットワーク設計問題に対して, タブー探索法を用いた解法を示してい

る．一つハブノードと一つ非ハブノードを交換する単一交換とハブノードと、ハブノードと非ハブノードの接続の変更のそれぞれに対して、短期メモリを設定している．

Campbell(1996) [13]

単一割当ハブネットワーク設計問題に対して、複数割当ハブネットワーク設計問題の解を用いて非ハブノードとハブノードの間のフロー量が最大となる接続を選択する最大フロー法、およびすべての組合せから最良の接続を求める割当フロー法を提案している．

Ernst-Krishnamoorthy(1996) [27]

単一ハブネットワーク設計問題に対して、アニーリング法を示している．局所探索では、交換、移動、特別移動の三つの近傍を利用している．ベンチマーク問題である CAB データに対して解法を適用している．

Ernst-Krishnamoorthy(1998) [28]

複数割当ハブネットワーク設計問題に対して、需要に関する強制制約式を含む強い定式化を示している．緩和解が満足しない強制制約式を追加する切除平面法を示している．

Geoffrion-Graves(1974) [35]

サプライチェーンネットワーク設計問題やロジスティクスネットワーク設計問題のさきがけの研究である．Hunt-Wesson Foods 社の事例を固定費用をもつ多品種フローモデルとして捕らえ、このモデルに Benders 分解原理を適用し、解を求めている．

Brown-Graves-Honczarenko(1987) [12]

NABISCO 社を対象とした長中期的な生産・配送ネットワーク設計問題を扱い、工場への生産施設の割当、生産する製品の割当、配送経路を決定する問題を固定費用をもつネットワーク設計問題として定式化している．この定式化に目標分解法を適用し、大規模な事例問題を効果的に解いている．

Arntzen-Brown-Harrion-Trafton(1995) [4]

生産、輸送、ベンダーを含むグローバルサプライチェーンネットワークモデルを用いて、DEC 社のネットワーク案を評価し、生産・輸送戦略案を決定している．このモデルは、グローバル、多製品、多段階、原料供給、在庫補充、輸送手段、税金、関税、関税還付、地域用件などさまざまな要因をふくむ大規模な混合整数計画問題である．

Goetschalckx-Vidal-Dogan(2002) [38]

中期的な生産割当、輸送割当と移転価格を扱ったサプライチェーンネットワーク設計モデルを対象としている．このモデルでは、供給者、生産工場、物流センター、顧客とその需要を既知として、税引き後利益が最大となる生産・輸送システムの構成と多数の支社間の移転価格を求めており、二つの事例を示している．

Santoso-Ahmed-Goetschalckx-Shapiro(2005) [70]

現実的なサプライチェーンネットワーク設計問題を解くために、確率的計画モデルとその解法を提案している。この解法は、標本平均近似法とよばれ、Benders 分解法を用いて膨大なシナリオをもつ確率的な設計問題を効率的に解く方法である。

2.1.5 サーベイ

ネットワーク設計問題の対しては多くのサーベイが示されている。ここでは、本論文に關係する容量制約のないネットワーク設計問題と容量制約をもつネットワーク設計問題を含むサーベイを示す。

Magnanti-Wong(1984) [59]

最小木問題，最短路問題，巡回セールスマン問題，施設配置問題，利用者均衡フロー問題について解説し，ネットワーク設計問題に対する Benders 分解法，分枝限定法，Lagrange 緩和法と線形緩和法，計算量とヒューリスティックを解説している。また，凸費用関数をもつネットワーク設計問題および凹費用関数をもつネットワーク設計問題も解説している。

Wong(1984) [75]

フロー費用がフロー量の線形である問題と非線形である問題を整理している。容量制約のない問題に対して，Benders 分解法における Benders カットを分類し，従来の研究を整理している。さらに，交通ネットワークに関する解法も整理している。

Wong(1985) [76]

最適化問題の文献目録集のネットワーク設計問題の章である。ネットワーク設計問題の一般モデル，スタイナー木問題および容量制約をもつ最小木問題に対する最適解法，近似解法などについて，従来の研究の要約を記述している。

Minoux(1989)[61]

ネットワーク設計問題のサーベイである。凹費用をもつフロー問題，ネットワーク設計問題，予算制約をもつ問題，多端末ネットワーク設計問題，容量制約をもつ最小木問題，階層ネットワーク設計問題，非同期ネットワーク設計問題などの問題を定義し，解法を解説している。

Balakrishnan-Magnanti-Mirchandani(1997) [6]

組合せ最適化問題の文献目録集のネットワーク設計問題の章である。容量制約のないネットワーク設計問題，容量制約をもつ最小木問題，容量制約をもつネットワーク設計問題，ネットワーク積込問題，ネットワーク修復問題，ネットワーク設計問題のバリエーション，および通信やサプライチェーンなどにおける応用問題などについて，従来の研究の要約を記述している。

Gendron-Crainic-Frangioni(1997)[34]

容量制約をもつネットワーク設計問題に関する定式化と解法のサーベイである。アークフローによる定式化をもとにした線形緩和による弱い緩和問題と強い緩和

問題を示し, Lagrange 緩和によるフロー緩和問題, 多品種フロー緩和問題, ナップサック緩和問題, フローナップサック緩和問題, 多品種フローナップサック緩和問題を示している. さらに, これらの最適値の関係を分析している. 100 ノード, 700 アーク, 400 品種までの問題について数値実験を行い, 分枝限定法による最適値と各種緩和法による下界値を比較している.

片山直登 (2002) [47]

交通ネットワークフロー問題, 予算制約をもつネットワーク設計問題, 容量制約のないネットワーク設計問題, 交通ネットワーク設計問題, 通信ネットワーク設計問題, 容量制約をもつ最小木問題, 容量制約をもつネットワーク設計問題など幅広いネットワーク設計問題を解説している. また, 定式化や計算複雑性などについても解説している.

Crainic(2003)[16]

長距離輸送計画問題のサーベイである. タクティカルレベル, ストラテジックレベル, オペレーショナルレベルに分けて, 計画問題を解説している. 基本的なネットワーク設計問題のモデルと解法を解説し, サービスネットワーク設計問題, 頻度保証ネットワーク設計問題, 空トラックフロー問題や割当問題などの Less-Than-Truckload 問題を解説している.

Costa(2005)[15]

Benders 分解法の始点から, ネットワーク設計問題の従来の研究を整理している. 容量制約のない設計問題, 容量制約をもつ設計問題, 階段状費用をもつ問題などに対する Benders 分解法を解説している.

片山直登 (2008) [48]

最短路問題, 最小木問題などのネットワーク問題, 最小費用フロー問題, 多品種フロー問題, 利用者均衡フロー問題などのネットワークフロー問題, 予算制約をもつネットワーク設計問題, 容量制約のないネットワーク設計問題, 容量制約をもつ最小木問題, 容量制約をもつネットワーク設計問題, ハブネットワーク設計問題などのネットワーク設計問題を解説している.

2.2 従来の研究と本論文の各章の位置づけ

従来の研究と本論文の各章の位置づけを表に示す. 表 2.1は, 対象とするモデル (単一品種と多品種) と研究内容 (解析, 近似解法, 緩和法, 最適解法) の項目で, 従来の研究を分類した一覧表である. 表 2.2は, 容量制約の有無, 区分的線形と研究内容 (解析, 近似解法, 緩和法, 最適解法) の項目で, 従来の研究を分類した一覧表であり, 手法を記載している. 二つの表の該当する分類に, 本論文の 3 章から 6 章を記載する.

第 3 章では, 多品種の容量制約のないネットワーク設計問題に対して, 強い妥当不等式を示し, 二種類の近似解法を提案する. 最適化問題に対しては, 解析的な面から強い妥当

不等式が必要とされている．強い妥当不等式を用いると最適解に近い緩和解が求められることができ，よい下界値を算出することができ，かつ緩和解を用いた近似解法が有効に機能する可能性がある．また，この容量制約のないネットワーク設計問題に対して，数多くの解法が提案されてきた．精度の高い解を算出する解法として Scott(1969) のバックワード法があるが，この解法が算出する解は精度が高いものの解法の計算量が大きく，大規模な問題への適用が困難である．また，大規模な問題に対して適切な解を算出できる高速な解法として Minoux(1976) の解法があるが，バックワード法と比べて算出する解の精度が劣るという欠点がある．一方，厳密な解を求めるために，最適化ソフトウェア用いて直接強い定式化を解く方法があるが，計算時間と定式化の規模によりある程度の規模までの問題にしか適用することができない．提案する強い妥当不等式は Balakrishnan(1987) の不等式を発展させたものである．提案する第一の近似解法は，Minoux(1976) の貪欲法を改良した方法である．この改良により，バックワード法に比べて計算時間を抑え，かつバックワード法に近い精度で解を算出することが可能となる．第二の近似解法は，列生成法と行生成法を用いた線形緩和法であり，容量制約のないネットワーク設計問題では新しい解法である．最適化ソフトウェア用いて，強い定式化を直接解く場合よりも，安定してよい解を算出でき，バックワード法よりも精度の高い解を算出することが可能となる．

第4章では，多品種の容量制約のないネットワーク設計問題に対して，三種類の近似解法を提案する．この問題に関連した問題に対する解法として，Holmberg(1998) の Lagrange 緩和法がある．この解法は向きのないモデルに対する解法であり，向きのあるモデルに対しては必ずしも有効でない場合がある．そこで，向きのあるモデルの性質を利用した強い妥当不等式を考慮することによって，より良い解を算出することが可能となる．提案する第一の近似解法は，Holmberg の Lagrange 緩和法を向きのないモデルに拡張したものである．第二の解法と第三の解法は，強い妥当不等式を含めた定式化を用いた解法である．なお，二番目の解法は Holmberg の Lagrange 緩和法よりも早い時期的に提案しており，また Holmberg は分枝限定法と組み合わせた解法も示している．

第5章では，多品種の容量制約をもつネットワーク設計問題に対して，二つの近似解法を提案する．この問題に対してメタ解法とよばれる解法が適用され，年々より精度の高い解を算出できる解法が開発されてきている．しかし，算出する解の精度は十分でない場合もあり，かつ多くの計算時間が必要とされている．提案する解法は，単一品種問題に対する Pochet(2004) の容量スケールリング法を多品種の問題の強い定式化に適用した方法，従来からある列生成法，さらに新しい考えである行生成法と分枝限定法を組み合わせた方法である．これにより，従来解法よりも精度が高い解を短時間で算出することが可能となる．提案する解法は，Crainic(2004) の傾斜スケールリング法が比較的内容が類似している．

第6章では，区分的線形費用をもつネットワーク設計問題に対して，二つの近似解法を提案する．この問題に対しては Croxton(2007) が示した閾値法があるが，問題の構造を解析する研究の中の一部として紹介されている方法であり，十分な議論や分析がなされている訳ではない．また，この解法は，問題によっては実行可能解を算出できないという欠点をもつ．提案する解法は，Croxton(2007) の三種類の定式化に容量スケールリング法を適

表 2.1 従来の研究と本論文の各章の位置づけ

	解析	近似解法	緩和法	最適解法
単 一 品 種	容量制約をもつネットワークフロー問題			
	–	Pochet(2004)	–	–
	区分的線形費用をもつネットワークフロー問題			
	Croxton(2003)	Kim(2000) Kim(2000) Kim(2006)	–	–
多 品 種	予算制約をもつネットワーク設計問題			
	Johnson(1978) Wong(1980) Wong(1985)	Scott(1969) Dionne(1979) Boffey(1979) 森津(1984)	Gallo(1983) Ahuja(1987) 片山(1995)	Scott(1969) Boyce(1973) Hoang(1973) Dionne(1979) Boffey(1979) 森津(1984)
	容量制約のないネットワーク設計問題			
	Balakrishnan(1987) 第3章	Billheimer(1973) Minoux(1976) Los(1982) 第3章	Balakrishnan(1989) Lamar(1990) Holmberg(1998) 第4章	Los(1982) Magnanti(1986) Guieřez(1996) Holmberg(1998)
	容量制約をもつネットワーク設計問題			
	Magnanti(1993) Barahona(1996) Chouman(2003)	Gendron(1996) Crainic(2000) Crainic(2002) Ghamlouche(2003) Crainic(2004) Ghamlouche(2004) Zaleta(2004) Álvarez(2005) Álvarez(2005) Crainic(2006) 第5章	片山(1993) Gendron(1994) Gendron(1994) Herrmann(1996) Holmberg(2000) Crainic(2001)	Holmberg(2000)
	区分的線形費用をもつネットワーク設計問題			
Muriel(2004) Croxtion(2007)	Croxton(2003) Muriel(2004) Croxtion(2007) 第6章	Croxton(2003)	–	

用した方法と、Croxtionの拡張した定式化をパスフローに直した定式化に容量スケール
ング法、行生成法と列生成法を適用した方法である。

表 2.2 多品種問題に対する従来の研究と本論文の各章の位置づけ

容量	分類	研究	手法
なし	解析	Balakrishnan(1987) 第3章	切除平面 妥当不等式
	近似解法	Billheimer(1973) Minoux(1976) Los(1982) 第3章	貪欲法 貪欲法 貪欲法 線形緩和法
	緩和法	Balakrishnan(1989) Lamar(1990) Holmberg(1998) 第4章	双対上昇法 容量改善法 Lagrange 緩和法 Lagrange 緩和法
	最適解法	Los(1982) Magnanti(1986) Holmberg(1998) Guieřrez(1996)	分枝限定法 Benders 分解法 分枝限定法 Benders 分解法
あり	解析	Magnanti(1993) Barahona(1996) Chouman(2003)	多面体解析, 妥当不等式 妥当不等式 妥当不等式
	近似解法	Gendron(1996) Crainic(2000) Crainic(2002) Ghamlouche(2003) Crainic(2004) Ghamlouche(2004) Zaleta(2004) Álvarez(2005) Álvarez(2005) Crainic(2006) 第5章	貪欲法 タブー探索法 タブー探索法 タブー探索法 傾斜スケーリング法 パス再結合法 タブー探索法 散布分散法 散布分散法 タブー探索法 容量スケーリング法
	緩和法	片山直登(1993) Gendron(1994) Gendron(1994) Herrmann(1996) Holmberg(2000) Crainic(2001)	双対上昇法 Lagrange 緩和法 Lagrange 緩和法 双対上昇法 Lagrange 緩和法 Lagrange 緩和法
	最適解法	Holmberg(2000)	分枝限定法
区的分的線形	解析	Muriel(2004) Croxtton(2007)	妥当不等式 妥当不等式
	近似解法	Croxtton(2003) Muriel(2004) Croxtton(2007) 第6章	分枝限定法 線形緩和ヒューリスティック法 閾値法 容量スケーリング法
	緩和法	Croxtton(2003)	切除平面法, 分枝限定法

第3章

容量制約のないネットワーク設計問題の近似解法

3.1 本章の概要

ネットワークを設計する際に考慮すべき基本的な費用は、アークを選択するときに発生する費用であるデザイン費用とモノがアーク上を移動するときに発生するフロー費用である。本章では、アーク容量の制限を考慮しない条件のもとで、これら二種類の費用の合計を最小にするような多品種のネットワーク設計問題を考える。アーク容量の制限を考慮しないため、この問題は容量制約のないネットワーク設計問題 (*UND*) とよばれる。

UND に対しては、今日までに高速な近似解法や精度の高い解を算出する近似解法が開発されている。高速な近似解法は算出する解の精度が悪い場合があり、その誤差は問題の規模にしたがって大きくなる傾向にある。高額なネットワーク設計の費用を考えると、その誤差による損失は大きなものとなる。また、従来の精度の高い解を算出する近似解法は大きな計算時間を必要とするため、実用規模の問題の解を実用的な時間では算出できないという状況も生じる可能性がある。このような点を踏まえ、*UND* に対して、従来の精度の高い解を算出する近似解法と同程度の誤差で、かつ計算時間を大幅に抑えることのできる近似解法の開発が必要である。またその一方で、計算時間はかかるものの、従来の精度の高い解を算出する近似解法よりもさらに精度の高い解を算出する近似解法が必要である。

本章では、新しい妥当不等式である非順流不等式を示し、*UND* に対する二種類の近似解法を提案する。第一の解法は、不要であろうアークを順次除いていく貪欲法である。この解法は、Minoux 法 [61] における目的関数値の変化量の評価を厳密化した方法であり、大規模な問題を高速に解くことができる解法である。

第二の解法は、強い妥当不等式を含む定式化の線形緩和解にもとづく解法である。これは、パスフロー変数を用いた定式化とパスフローに関する列生成法と行生成法を用いた線形緩和法であり、より高精度の解を求めることができる。列生成法は、変数を限定した問題からはじめ、逐次、基底解となる変数を生成する方法であり、多くの研究で成果を挙げ

ている手法である．また，行生成法は，逐次，列生成法で生成された変数がはじめて含まれる制約式を生成する方法である．類似した手法として切除平面法があり，これは緩和解が満たさない制約式を，逐次，追加する手法である．

UND の前提条件，使用する記号および UND の定義，アークフローを用いた定式化とパスフローを用いた定式化，および UND に対する強い妥当不等式を示す．続いて，Minoux 法の改良法および列生成法と行生成法を示し，提案した解法の有効性を示すために数値実験を行い，その結果を示す．最後に，本章のまとめを記述する．

3.2 問題の定式化

はじめに， UND の前提条件，使用する記号と UND の定義を示す．続いて，アークフローを用いた定式化とパスフローを用いた定式化，および UND に対する強い妥当不等式を示す．

3.2.1 前提条件，記号および問題の定義

前提条件

- ノード集合が与えられている．
- アーク集合が与えられている．
- アークは向きをもたない．
- アークの処理量には，容量である上限を考慮しない．
- アークには，非負のデザイン費用が与えられている．
- アークには，非負の単位当たりのフロー費用が与えられている．
- 複数の品種からなる品種集合が与えられている．
- 各品種の需要は 1 である．
- 各品種の需要は，始点から終点までのパス上を移動する．

この前提では，需要をすべて 1 として基準化している．実際の需要が 1 ではない場合には，フロー費用を需要で割ったものを新たな単位当たりのフロー費用とし，需要を 1 とすれば，需要を 1 に基準化した問題に置き換えることができる．

記号の定義

- N : ノード集合
- A : アーク集合
- K : 品種集合
- N_n : ノード n に接続するアークの他方の端点の集合
- P^k : 品種 k の取り得るパス集合
- K_{ij} : アーク (i, j) 上を移動している品種の集合
- K_n^o : ノード n を始点とする品種の集合

- K_n^d : ノード n を終点とする品種の集合
- O^k : 品種 k の始点
- D^k : 品種 k の終点
- $K_{O^k}^o$: 品種 k の始点 O^k を始点とする品種の集合
- $K_{D^k}^d$: 品種 k の終点 D^k を終点とする品種の集合
- $K_{O^k}^d$: 品種 k の始点 O^k を終点とする品種の集合
- \bar{P}^k : 品種 k の取り得るパス集合の部分集合
- \bar{P} : \bar{P}^k の $k(\in K)$ に関する和集合
- $A_{\bar{P}^k}$: \bar{P}^k に含まれるパスに含まれるアークの集合
- $A_{\bar{P}}^{kh}$: \bar{P} に含まれる品種 k のパスと品種 h のパスに共通に含まれるアークの集合
- c_{ij}^k : アーク (i, j) における品種 k の単位当りの非負のフロー費用
- f_{ij} : アーク (i, j) の非負のデザイン費用
- δ_{ij}^p : パス p にアーク (i, j) が $i \rightarrow j$ 向きに含まれるとき 1, そうでないとき 0 を表す定数
- x_{ij}^k : 品種 k のフローがアーク (i, j) 上を $i \rightarrow j$ 向きに移動する量を表すアークフロー変数 ; 非負の連続変数
- z_p^k : 品種 k のフローがパス p 上を移動する量を表すパスフロー変数 ; 非負の連続変数
- y_{ij} : アーク (i, j) を選択するとき 1, そうでないとき 0 であるデザイン変数 ; 0-1 変数
- π^k : 品種 k の需要保存式に対する双対変数 ; 非負の連続変数
- w_{ij}^k : アーク (i, j) における品種 k の需要に関する強制制約式に対する双対変数 ; 非負の連続変数
- s_{ij}^{kh} : アーク (i, j) における始点を同一にする品種 k と h に関する非逆流不等式に対する双対変数 ; 非負の連続変数
- t_{ij}^{kh} : アーク (i, j) における終点を同一にする品種 k と h に関する非逆流不等式に対する双対変数 ; 非負の連続変数
- u_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $i \rightarrow j$ 向きの非順流不等式に対する双対変数 ; 非負の連続変数
- r_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $j \rightarrow i$ 向きの非順流不等式に対する双対変数 ; 非負の連続変数

問題の定義

ノード集合 N , デザイン費用 f とフロー費用 c をもつ向きをもたないアーク集合 A , 品種の需要 1 をもつ品種集合 K が与えられている. このとき, フロー費用とデザイン費用の合計を最小にするアーク集合 $A'(\subseteq A)$ とフロー x または z を求めよ.

3.2.2 アークフローを用いた定式化

UND のアークフローによる定式化 UAF は次のようになる。

(UAF)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.1)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (3.2)$$

$$x_{ij}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.3)$$

$$x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.4)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (3.5)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3.6)$$

(3.1) 式は目的関数であり，フロー費用とデザイン費用の総和を最小化する．(3.2) 式はフロー保存式である．このフロー保存式は，ノードに流入するフローと流出するフローの差が，品種 k の始点であれば -1 ，終点であれば 1 ，その他のノードであれば 0 であることを表す．この式は，各品種について，始点から終点まで需要が必ず移動することを保証する．(3.3)，(3.4) 式は， x_{ij}^k, x_{ji}^k の上限が 1 であることを考慮したアークにおける品種の需要に関する強制制約式である．アーク (i, j) が選択されたときには品種 k のアーク上のフロー量が最大で 1 となり，アーク (i, j) が選択されないときにはフロー量が 0 となることを表す．(3.5) 式はアークフロー変数の非負制約であり，(3.6) 式はデザイン変数の $0-1$ 条件である． UAF は， $|A||K|$ 個のアークフロー変数， $|A|$ 個のデザイン変数，および $|N||K| + 2|A||K|$ 本の制約式をもつ問題となる．

図 3.1(a) のように，同一品種 k のフローが同一のアーク上で反対方向に移動するフローが存在するものとする．フロー費用が正であれば，図 3.1(b) に示すように反対方向を移動しない実行可能で，かつフロー費用が安いフローが存在する．また，フロー費用が 0 であれば，同じ費用のフローが存在する．このため，最適解のひとつにおいては，同一品種 k のフローは同一のアーク上で反対方向に移動しないことになり， $x_{ij}^k \leq y_{ij}$ または $x_{ji}^k \leq y_{ij}$ のどちらか一方が成り立つことから，次式は妥当不等式となる．

$$x_{ij}^k + x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.7)$$

この式は，(3.3) 式または (3.4) 式を持ち上げた式となる．

品種について (3.7) 式の和を取ると，次式が成り立つ．

$$\sum_{k \in K} (x_{ij}^k + x_{ji}^k) \leq |K| y_{ij} \quad \forall (i, j) \in A \quad (3.8)$$

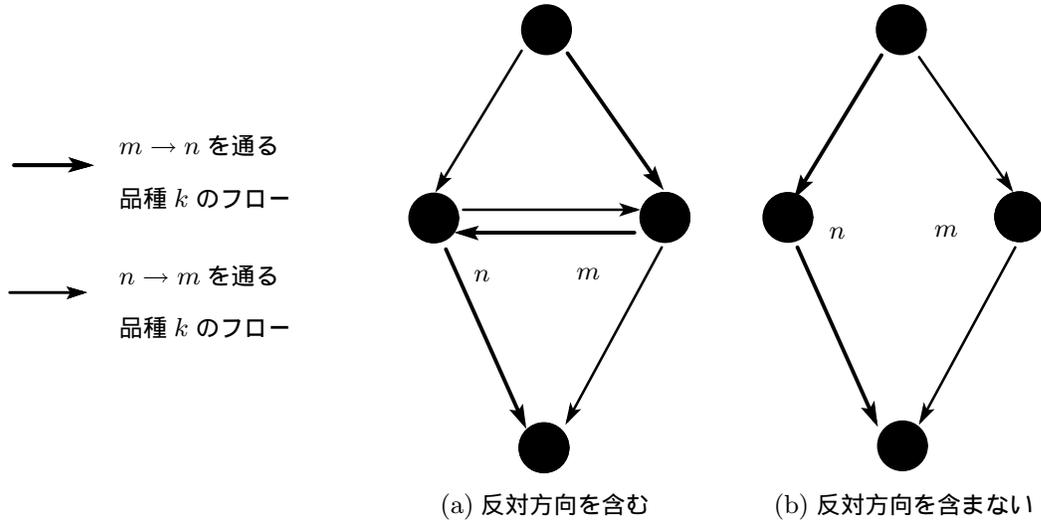


図 3.1 非集約化した強制制約式

この式は，アーク (i, j) が選択されたときにはこのアーク上のフロー量が最大で $|K|$ となり，アーク (i, j) が選択されないときにはこのアーク上のフロー量が 0 となることを表す．これは，フロー変数とデザイン変数の関係を表すアークにおける強制制約式であり，(3.7) 式を集約化した式となる．一方，(3.7) 式は (3.8) 式を品種毎に非集約化した式となる．

3.2.3 パスフローを用いた定式化

パスフロー変数を用いて， UND の定式化を行うことができる．パスフロー変数を用いた定式化 UPF は次のようになる．

(UPF)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.9)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad \forall k \in K \quad (3.10)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.11)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.12)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (3.13)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3.14)$$

(3.9) 式は目的関数であり，フロー費用とデザイン費用の総和を最小化する．ここで， $\sum_{p \in P^k} \delta_{ij}^p z_p^k$ はアーク (i, j) を通るパスフローの合計となり x_{ij}^k に一致し， $\sum_{p \in P^k} \delta_{ji}^p z_p^k$ は x_{ji}^k に一致する．(3.10) 式は需要保存式であり，品種 k のパスフローの合計が品種 k の需要である 1 に一致することを表す．(3.11) 式は (3.3) 式に，(3.12) 式は (3.4) 式に対応

する強制制約式である。(3.13)式はパスフロー変数の非負制約であり,(3.14)式はデザイン変数の0-1条件である。

P^k の要素数はアーク数 $|A|$ に対して指数個となり,パスフロー変数の数は $\sum_{k \in K} |P^k|$ 個であり,指数個となる。 UPF は $\sum_{k \in K} |P^k|$ 個のパスフロー変数, $|A|$ 個のデザイン変数と, $|K| + 2|A||K|$ 本の制約式をもつ問題となる。変数が指数個と膨大なものとなるので,小規模な問題であってもこの定式化を直接解くことは困難である。実際には,逐次,必要なパスフロー変数を生成して問題を解く列生成法が用いられる。この列生成法をうまく適用すれば,アークフローによる定式化の場合よりも,陽的に使用する変数の数を抑えることができる。

3.3 強い妥当不等式

一般的に,問題の定式化に有効な妥当不等式を追加すると実行可能領域が狭くなり,緩和問題から得られる下界値が改善されるため,緩和問題をもとにして得られる近似解も改善される場合が多い。また,分枝限定法などでは計算時間が短縮できる可能性がある。この節では, UND に対する強い妥当不等式であるいくつかの強制制約式を示す。なお,ここでは始点(または終点)が同一ノード n である品種に対して,同一方向および反対方向のフロー費用が等しいか比例する場合を想定する。

3.3.1 非逆流不等式

図3.2(a)に示すように,始点(または終点)が同一ノードである品種 k, h のフローが,同一のアーク上で反対方向に移動するものとする。この場合,図3.2(b)に示すように,反対方向に移動しないような実行可能で,かつフロー費用がより安い(フロー費用が0であれば同じ費用の)フローが存在する。このため,終点(または始点)が同一ノードである品種のフローが同一のアーク上で反対方向に移動しない最適解が存在する。このことから,Balakrishnan-Magnanti-Wong[7]は,次のような妥当な強制制約式である非逆流不等式を示している。

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (3.15)$$

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (3.16)$$

ここで, $K_{O^k}^o$ は k の始点 O^k を始点とする品種の集合, $K_{D^k}^d$ は k の終点 D^k を終点とする品種の集合である,

(3.15)式において,左辺は品種 k の $i \rightarrow j$ 向きのフローと k と始点を同じにする品種 h の $j \rightarrow i$ 向きのフロー量の和であり,これがアーク (i, j) が選択されたときに最大で1,選択されないときに0となることを表す。同様に,(3.16)式において,左辺は品種 k の $i \rightarrow j$ 向きのフローと k と終点を同じにする品種 h の $j \rightarrow i$ 向きのフロー量の和であり,

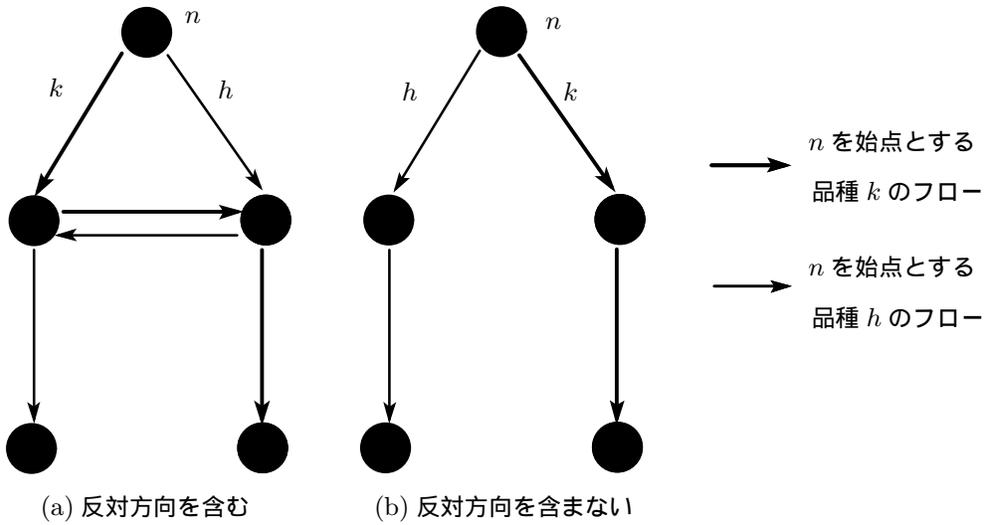


図 3.2 非逆流不等式

これがアーク (i, j) が選択されたときに最大で 1, 選択されないときに 0 となることを表す. これらの非逆流不等式の数は, $|A||K|^2/|N|$ 本となる.

x_{ij}^k, x_{ji}^h が 0-1 変数であれば, すべての反対方向の移動を排除できる. しかし, 連続変数であるため, 例えば $x_{ij}^k = x_{ji}^h = 0.5$ がこれらの制約式を満足するように, すべての反対方向に移動するフローを排除できる訳ではない.

パスフロー変数を用いると, これらの非逆流不等式は次のようになる.

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (3.17)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (3.18)$$

3.3.2 非順流不等式

図 3.3(a) に示すように, 同一ノード n を始点とする品種 k と終点とする品種 h のフローが, 同一のアーク上で同一方向に移動するフローが存在するものとする. この場合, 図 3.3(b) に示すように, 同一方向に移動しないような実行可能で, かつフロー費用が安い (フロー費用が 0 であれば同じ費用の) フローが存在する. このため, 同一ノードを始点とする品種と終点とする品種のフローが同一のアーク上で同一方向に移動しない最適解が存在する. したがって, 次の非順流不等式は妥当な不等式となる.

$$x_{ij}^k + x_{ij}^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.19)$$

$$x_{ji}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (3.20)$$

ここで, $K_{O^k}^d$ は品種 k の始点 O^k を終点とする品種の集合である.

(3.19) 式において, 左辺は品種 k の $i \rightarrow j$ 向きのフローと k の始点を終点とする品種 h の $i \rightarrow j$ 向きのフロー量の和であり, これがアーク (i, j) が選択されたときに最大で 1,

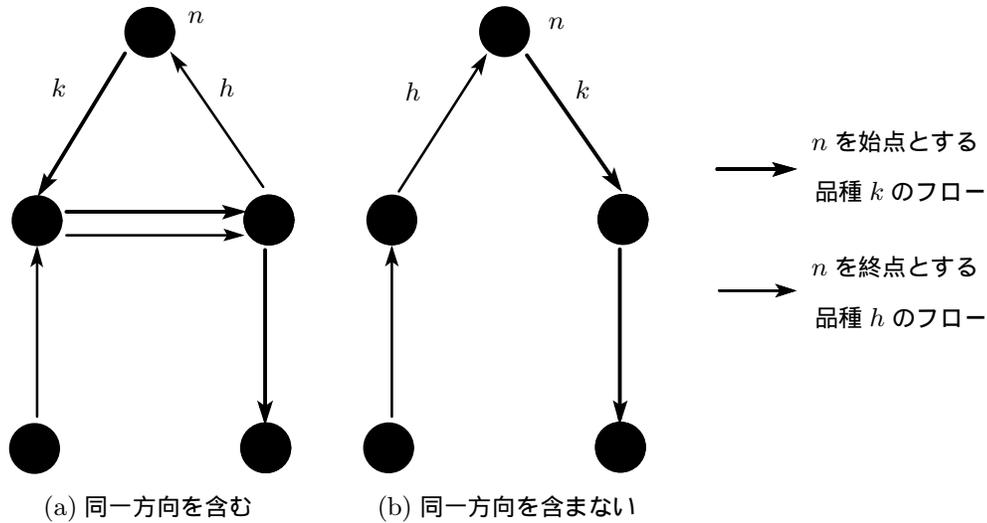


図 3.3 非順流不等式

選択されないときに 0 となることを表す．同様に，(3.20) 式において，左辺は品種 k の $j \rightarrow i$ 向きのフローと k の始点を終点とする品種 h の $j \rightarrow i$ 向きのフロー量の和であり，これがアーク (i, j) が選択されたときに最大で 1，選択されないときに 0 となることを表す．これらの非逆流不等式の数， $|A||K|^2/|N|$ 本となる．

非逆流不等式の場合と同様に， x_{ij}^k, x_{ji}^h が 0-1 変数であれば，すべての同一方向のフローを排除できる．しかし，フロー変数が小数値をとる場合もあるため，すべての同一方向のフローを排除できる訳ではない．

パスフロー変数を用いると，これらの反順流不等式は次のようになる．

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.21)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.22)$$

3.4 Minoux 法の改良法

この節では，貪欲法である Minoux 法を示した後，Minoux 法における目的関数値の変化量の評価を厳密にした Minoux 法の二種類の改良法を示す．

3.4.1 Minoux 法

UND に対する貪欲法として，ネットワークから不要であろうと思われるアークを順々に取り除いていくバックワード法 [71] がある．取り除くアークは，現在のネットワーク上で，取り除いたときに最も目的関数が減少するアークである．アークを取り除くたびにネットワークの形状が変化するため，バックワード法では，現在のネットワークに含まれているアークをそれぞれ除いたときの目的関数値の減少量を再評価する必要がある．

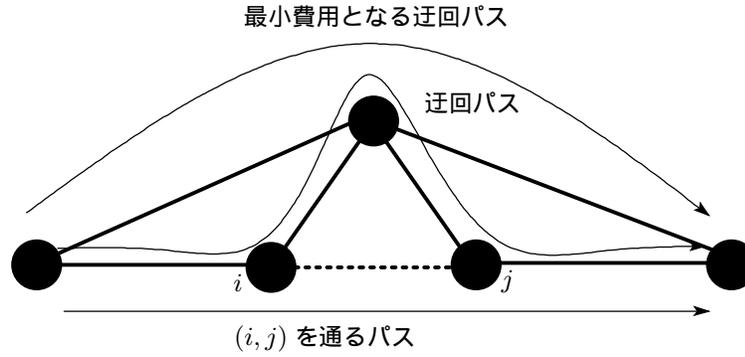


図 3.4 目的関数値の減少量

Warshall-Floyd 法を用いて減少量を厳密に評価すると $O(|N|^3)$ の計算量を必要とするため、バックワード法の計算量は $O(|A|^2|N|^3)$ となる。このため、アーク数が多いネットワークでは膨大な計算量が必要となる。

そこで、アークを取り除くたびにすべてのアークに対して目的関数値の減少量を厳密に再評価するのではなく、最も減少量大きい可能性のあるアークのみに対して減少量を再評価することによって、計算量を抑える Minoux 法 [61] が示されている。

はじめに、アーク $(i, j) \in A$ に対してこのアークを取り除いたときのノード i, j 間のフロー費用が最小となるパスに、現在、アーク (i, j) を移動するフローを迂回させたときのフロー費用の増加量を α_{ij} とする (図 3.4)。アーク (i, j) を取り除いたときの目的関数値の減少量の評価値を Δ_{ij} とし、次のようにおく。

$$\Delta_{ij} = f_{ij} - \alpha_{ij} \quad (3.23)$$

図 3.4 に示すように、アーク (i, j) の両端点を始点・終点としない品種では、アーク (i, j) を取り除いたときに i, j を経由しない最小費用となる迂回パスが存在する可能性がある。このため、実際のフロー費用の増加量は Δ_{ij} 以下となり、 Δ_{ij} は目的関数値の減少量の下限値となる。

一方、すべてのアークを含むネットワークから始め、アークを一本ずつ順々に取り除く操作を行うものとする。 l 回目の操作時のネットワークにおいて、ネットワークに含まれているアーク (i, j) を取り除いたときの目的関数値の減少量の評価値を Δ_{ij}^l とする。 l 回目と $l+1$ 回目の繰り返しにおけるネットワークに、アーク (i, j) が共に含まれているものとする。 l 回目と比べて $l+1$ 回目の方はアークが一本少ないことから、アーク (i, j) を $l+1$ 回目に取り除いたときのフロー費用の増加量は、 l 回目に取り除いたときの増加量以上となり、

$$\Delta_{ij}^{l+1} \leq \Delta_{ij}^l \quad (3.24)$$

が成り立つ。このため、 l 回目において Δ_{ij}^l が 0 以下であれば、それ以降も 0 以下となる。 $\Delta_{ij}^l \leq 0$ であるアークを取り除くと目的関数値は減少しないものと評価できる。このため、以後、このようなアークは取り除く対象から除外できる。

一方、バックワード法では、アークを削除する際に、現在のネットワークに含まれているアークをそれぞれ取り除いたときの目的関数の変化量を求める必要がある。しかし、 Δ_{ij}^l が大きなアークは、 Δ_{ij}^{l+1} も大きな値となると考えられる。そこで、 Δ_{ij}^l の降順に該当するアークと減少量をリストに格納しておく。 $l+1$ 回目において、 Δ_{ij}^l が最大のアーク、すなわちリストの先頭のアークについて、現在のネットワークにおける評価値 Δ_{ij}^{l+1} を再計算する。この減少量がリストの中で最大であるときにのみ、このアークをネットワークから取り除く。この減少量が最大とならないときには、このアークと減少量をリスト中の降順の位置に格納し、次に評価値の大きなアークについて、同様な操作を行う。多くの場合、数回の操作で削除すべきアークが見つかり、一本のアークを削除する際に $|A|$ 回の評価を行う必要がないため、高速な解法となる。

これらの性質を利用して、バックワード法を効率化した解法が Minoux 法である。

Minoux 法

- [ステップ 1] アーク集合 A で構成されるネットワークを初期ネットワークとする。
- [ステップ 2] ネットワークに含まれているすべてのアーク (i, j) について、このアークを取り除いた場合の目的関数値の減少量の評価値 Δ_{ij} を計算する。
- [ステップ 3] $\Delta_{ij} > 0$ であるすべてのアーク (i, j) について、 Δ_{ij} の降順に該当するアークと Δ_{ij} をリストに格納する。
- [ステップ 4] リスト内に $\Delta_{ij} > 0$ であるアーク (i, j) がなければ、終了する。そうでなければ、リストから Δ_{ij} が最大であるアークを取り出す。このアークを (p, q) とする。
- [ステップ 5] アーク (p, q) に対して、現在のネットワークにおける Δ_{pq} を再計算する。
- $\Delta_{pq} > 0$ で、リスト内で最大であれば、ステップ 6 へ。
 - $\Delta_{pq} > 0$ であれば、 Δ_{pq} の値にしたがって、アーク (p, q) と Δ_{pq} をリスト内の適切な位置に挿入し、ステップ 4 へ戻る。
 - $\Delta_{pq} \leq 0$ であれば、ステップ 4 へ戻る。
- [ステップ 6] アーク (p, q) をネットワークから取り除き、ステップ 4 へ戻る。

初期計算を除けば、Minoux 法はアークが実際に取り除く対象となる場合に限り減少量の近似的な評価値を計算するだけであるため、非常に高速な解法となる。Minoux 法の計算量は $O(|A|^2|N|^2)$ となるが、平均的には $O(|A||N|^2)$ であり、計算量も少なく、得られる近似解も比較的優れたものである。

3.4.2 Minoux 法の改良法

Minoux 法のステップ 2 およびステップ 5 では、削除したアークの両端点間の費用が最小となるパス上にこのアーク上のフローを迂回させたときのフロー費用の増加量をもとに、評価値を算出している。しかし図 3.4 に示すように、削除したアークの両端点間を経由して両端点間の迂回パス上を移動する場合よりも、このアークの両端点を経由しないパ

スを移動するフロー費用の方が安くなる場合がある．このような場合，ステップ 2 およびステップ 5 で求められた目的関数値の減少量は，この時点でのアークに対する目的関数値の減少量の厳密な評価値ではなく，下界値となる．また， $\Delta_{ij} \leq 0$ であるアークであっても，実際に取り除いたときの目的関数値の減少量が正となる可能性もある．

このように，目的関数値の減少量の評価値に下界値を用いているのは，計算量の増加を防ぐためである．しかし，アーク (i, j) 上を通る品種数がそれほど多くない場合には，この評価を厳密に行っても，それほど計算量が増加しない．この厳密な評価値は，Murchland 法 [63, 73] によって効率的に求めることができる．

そこで，目的関数値の減少量の評価に厳密性を加えた二つのアルゴリズムを提案する．現在のネットワークにおいて，アーク (i, j) に対してこのアークを取り除いたとき，アーク (i, j) 上を移動している品種 k に対して， O^k, D^k 間の最小フロー費用パスを求め，このパスに品種 k のフローを迂回させたときのフロー費用の増加量を β_{ij}^k とする．アーク (i, j) を取り除いたときの目的関数値の減少量を $\hat{\Delta}_{ij}$ とし，次のようにおく．

$$\hat{\Delta}_{ij} = f_{ij} - \sum_{k \in K_{ij}} \beta_{ij}^k \quad (3.25)$$

とする．ここで， K_{ij} はアーク (i, j) 上を移動している品種集合である． Δ_{ij} はアーク (i, j) を取り除いたときの目的関数の減少量の下界値であるが， $\hat{\Delta}_{ij}$ は厳密な減少量となる．

改良法 1 は，Minoux 法のステップ 5 において目的関数の減少量を厳密に計算する修正を行ったものである．また，改良法 2 は，さらにステップ 2 において同様な修正を行ったものである．

改良法 1

[ステップ 1] アーク集合 A で構成されるネットワークを初期ネットワークとする．

[ステップ 2] ネットワークに含まれているすべてのアーク (i, j) について，このアークを取り除いた場合の目的関数値の減少量の評価値 Δ_{ij} を計算する．

[ステップ 3] $\Delta_{ij} > 0$ であるすべてのアーク (i, j) について， Δ_{ij} の降順に該当するアークと Δ_{ij} をリストに格納する．

[ステップ 4] リスト内に $\Delta_{ij} > 0$ であるアーク (i, j) がなければ，終了する．そうでなければ，リストから Δ_{ij} が最大であるアークを取り出す．このアークを (p, q) とする．

[ステップ 5] アーク (p, q) に対して，現在のネットワークにおける $\hat{\Delta}_{pq}$ を計算する． $\Delta_{pq} := \hat{\Delta}_{pq}$ とする．

- $\Delta_{pq} > 0$ で，リスト内で最大であれば，ステップ 6 へ．
- $\Delta_{pq} > 0$ であれば， Δ_{pq} の値にしたがって，アーク (p, q) と Δ_{pq} をリスト内の適切な位置に挿入し，ステップ 4 へ戻る．
- $\Delta_{pq} \leq 0$ であれば，ステップ 4 へ戻る．

[ステップ6] アーク (p, q) をネットワークから取り除き，ステップ4へ戻る．

改良法2

[ステップ1] アーク集合 A で構成されるネットワークを初期ネットワークとする．

[ステップ2] ネットワークに含まれているすべてのアーク (i, j) について，このアークを取り除いた場合の目的関数値の減少量 $\hat{\Delta}_{ij}$ を計算する．

[ステップ3] $\hat{\Delta}_{ij} > 0$ であるすべてのアーク (i, j) について， $\hat{\Delta}_{ij}$ の降順に該当するアークと $\hat{\Delta}_{ij}$ をリストに格納する．

[ステップ4] リスト内に $\hat{\Delta}_{ij} > 0$ であるアーク (i, j) がなければ，終了する．そうでなければ，リストから $\hat{\Delta}_{ij}$ が最大であるアークを取り出す．このアークを (p, q) とする．

[ステップ5] アーク (p, q) に対して，現在のネットワークにおける $\hat{\Delta}_{pq}$ を再計算する．

- $\hat{\Delta}_{pq} > 0$ で，リスト内で最大であれば，ステップ6へ．
- $\hat{\Delta}_{pq} > 0$ であれば， $\hat{\Delta}_{pq}$ の値にしたがって，アーク (p, q) と $\hat{\Delta}_{pq}$ をリスト内の適切な位置に挿入し，ステップ4へ戻る．
- $\hat{\Delta}_{pq} \leq 0$ であれば，ステップ4へ戻る．

[ステップ6] アーク (p, q) をネットワークから取り除き，ステップ4へ戻る．

改良法1の計算量は， $O(|A|^2|N|^3)$ であるが，平均的には $O(|A||N|^3)$ である．また，改良法2の計算量も $O(|A|^2|N|^3)$ ，平均的には $O(|A||N|^3)$ である．

バックワード法は良い近似解を得ることができるが，計算量は平均的にも $O(|A|^2|N|^3)$ である．一方，Minoux法の改良法1および改良法2はバックワード法と同程度の解が得られ，計算時間は大幅に短縮できることが期待できる．

3.5 線形緩和法

強い妥当不等式を含むパスフロー変数を用いた定式化に対して，列生成法と行生成法を組み合わせた線形緩和問題の解法を提案する．さらに緩和解をもとに近似解を算出する解法である線形緩和法を提案する．

3.5.1 線形緩和問題と限定主問題

UND は0-1のデザイン変数を含む組合せ最適化問題であるため，デザイン変数の0-1条件を0から1の連続条件に緩和した線形問題を考え，この緩和問題を解くことにする．緩和問題の最適値は UND の最適値ではなく下界値となる．一般的には，線形緩和問題の最適解は小数値を含むため， UND の実行可能解とはならない．しかし，強い妥当不等式を含む場合，0-1変数の線形緩和解が0または1により近い緩和解が求められる可能性があり，これらの線形緩和解をもとに0-1解を導出する．

強い妥当不等式を含むパスフロー変数を用いた定式化 $UPFS$ は次のようになる .

($UPFS$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.26)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad \forall k \in K \quad (3.27)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.28)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.29)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (3.30)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (3.31)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.32)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.33)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (3.34)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3.35)$$

(3.26) ~ (3.29) , (3.34) , (3.35) 式は UPF と同一である . (3.30) と (3.31) は非逆流不等式 , (3.32) と (3.33) 式は非順流不等式である .

続いて , $UPFS$ における (3.35) 式を線形緩和した問題 $UPFSL$ を考える .

(UPFSL)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.36)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad \forall k \in K \quad (3.37)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.38)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.39)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (3.40)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (3.41)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.42)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k + \sum_{p \in P^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (3.43)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (3.44)$$

$$0 \leq y_{ij} \leq 1 \quad \forall (i, j) \in A \quad (3.45)$$

UPFSL は線形計画問題となるため、UPFS よりも解き易い問題となる。しかし、UPFSL には指数個のパスフロー変数を含むため、直接、この定式化を解くことは得策ではない。そこで、すべてのパスを対象とするのではなく、パス集合とパスフロー変数を限定した限定主問題を解き、逐次、必要なパスを生成し、これらのパスフロー変数を限定主問題に追加していく。生成するパスフロー変数が単体法の列に相当することから、このような方法を列生成法とよぶ。

一方、非逆流不等式と非順流不等式を含めため、制約式の数膨大なものとなる。パスフロー変数を限定した場合、これらの変数を含まない強制制約式が多数存在する。そのような制約式は、限定主問題では $0 \leq y_{ij}$ と左辺が定数 0 と表わされ、これらは不要な制約式となる。そこで、現在の限定主問題に含まれるパスフロー変数を含む強制制約式のみを制約式として使用する。列生成法によりパスフロー変数が生成されたときに、逐次、左辺のパスフロー変数が初めて生成された強制制約式を限定主問題に追加する。このような方法を行生成法とよぶ。

ここで、品種 $k \in K$ の適当なパスの初期集合 \bar{P}^k が求められているものとする。このとき、パス集合が $\bar{P} = (\bar{P}^k)$ に制限されている次のような限定主問題 UPFSLR(\bar{P}) を考える。

(UPFSLR(\bar{P}))

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in \bar{P}^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.46)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = 1 \quad \forall k \in K \quad (3.47)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}^k}, k \in K \quad (3.48)$$

$$\sum_{p \in \bar{P}^k} \delta_{ji}^p z_p^k \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}^k}, k \in K \quad (3.49)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}}^{kh}, h \in K_{O^k}^o, k \in K \quad (3.50)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}}^{kh}, h \in K_{D^k}^d, k \in K \quad (3.51)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.52)$$

$$\sum_{p \in \bar{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\bar{P}}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.53)$$

$$z_p^k \geq 0 \quad \forall k \in K, p \in \bar{P}^k \quad (3.54)$$

$$0 \leq y_{ij} \leq 1 \quad \forall (i,j) \in A \quad (3.55)$$

ここで, $A_{\bar{P}^k}$ は \bar{P}^k に含まれる品種 k のパスに含まれるアーク集合, $A_{\bar{P}}^{kh}$ は \bar{P} に含まれる品種 k と品種 h のパスに共通に含まれるアーク集合である. そのため, (3.48), (3.49) 式は, (i,j) を含む品種 k のパスのパスフロー変数が生成されているときのみ存在する強制制約式となる. また, (3.50) ~ (3.53) 式は, アーク (i,j) を含む品種 k のパスフロー変数と品種 h のパスフロー変数が共に生成されているときのみ存在する強制制約式となる.

この問題は線形計画問題であるため, パス集合の要素数が少なければ, 汎用の数理最適化ソフトウェアを用いて解くことができる.

3.5.2 列生成法と行生成法

UPFSLR(\bar{P}) はパスフロー変数が限定された問題であるため, その最適値は UPFSLR の最適値ではなく, 上界値となる. UPFSLR の最適値を求めるためには, 基底変換により目的関数値を減少できるようなパスおよびパスフロー変数を生成していく必要がある. 生成すべきパスフロー変数は, UPFSLR(\bar{P}) の最適解において被約費用が負であるパスフロー変数である.

UPFSLR において, パスフロー変数に対する被約費用を求めておく. 被約費用が負である変数がなければ, 最適解が得られたことになる. (3.37) 式に対する双対変数を π , (3.38) および (3.39) 式に対する非負の双対変数を w , (3.40) ~ (3.43) 式に対する非負の双対変数を s, t, u, r として, パスフロー変数 z に関する Lagrange 双対関数 $LF(z)$ を

作成する .

$$\begin{aligned}
LF(z) &= \sum_{(i,j) \in A} \sum_{k \in K} \left\{ \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ij}^p w_{ij}^k + \delta_{ji}^p c_{ji}^k + \delta_{ji}^p w_{ji}^k) z_p^k \right. \\
&\quad + \sum_{h \in K_{O_k}^o} \left(\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) s_{ij}^{kh} + \sum_{h \in K_{D_k}^d} \left(\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) t_{ij}^{kh} \\
&\quad \left. + \sum_{h \in K_{O_k}^d} \left(\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) u_{ij}^{kh} + \sum_{h \in K_{O_k}^o} \left(\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) r_{ij}^{kh} \right\} \\
&\quad - \sum_{k \in K} \sum_{p \in P^k} z_p^k \pi^k \\
&= \sum_{(i,j) \in A} \left[\sum_{k \in K} \sum_{p \in P^k} \{ \delta_{ij}^p (c_{ij}^k + w_{ij}^k) + \delta_{ji}^p (c_{ji}^k + w_{ji}^k) \} z_p^k \right. \\
&\quad + \sum_{k \in K} \left\{ \sum_{h \in K_{O_k}^o} \sum_{p \in P^k} \delta_{ij}^p z_p^k s_{ij}^{kh} + \sum_{h \in K_{D_k}^d} \sum_{p \in P^k} \delta_{ij}^p z_p^k t_{ij}^{kh} \right. \\
&\quad \left. + \sum_{h \in K_{O_k}^d} \sum_{p \in P^k} \delta_{ij}^p z_p^k u_{ij}^{kh} + \sum_{h \in K_{O_k}^o} \sum_{p \in P^k} \delta_{ji}^p z_p^k r_{ij}^{kh} \right\} \\
&\quad + \sum_{h \in K} \left\{ \sum_{k \in K_{O_h}^o} \sum_{p \in P^h} \delta_{ji}^p z_p^k s_{ij}^{hk} + \sum_{k \in K_{D_h}^d} \sum_{p \in P^h} \delta_{ji}^p z_p^k t_{ij}^{hk} \right. \\
&\quad \left. + \sum_{k \in K_{O_h}^d} \sum_{p \in P^h} \delta_{ij}^p z_p^k u_{ij}^{hk} + \sum_{k \in K_{O_h}^o} \sum_{p \in P^h} \delta_{ji}^p z_p^k r_{ij}^{hk} \right\} \Big] \\
&\quad - \sum_{k \in K} \sum_{p \in P^k} z_p^k \pi^k
\end{aligned}$$

ここで, $h \in K, k \in K_{O_h}^o$ は始点が同一である品種対の集合の要素を表しているの
で, $k \in K, h \in K_{O_k}^o$ に置き換えることができる. 同様に, $h \in K, k \in K_{D_h}^d$ は
 $k \in K, h \in K_{D_k}^d$ に置き換えることができる. また, $h \in K, k \in K_{O_h}^d$ は始点を終点とす
る品種対の集合の要素を表しているの, $k \in K, h \in K_{D_k}^o$ に置き換えることができる.

$$\begin{aligned}
LF(z) &= \sum_{(i,j) \in A} \left[\sum_{k \in K} \sum_{p \in P^k} \{ \delta_{ij}^p (c_{ij}^k + w_{ij}^k) + \delta_{ji}^p (c_{ji}^k + w_{ji}^k) \} z_p^k \right. \\
&\quad + \sum_{k \in K} \sum_{p \in P^k} \left\{ \delta_{ij}^p \left(\sum_{h \in K_{O_k}^o} s_{ij}^{kh} + \sum_{h \in K_{D_k}^d} t_{ij}^{kh} + \sum_{h \in K_{O_k}^d} u_{ij}^{kh} + \sum_{h \in K_{D_k}^o} r_{ij}^{kh} \right) \right. \\
&\quad \left. + \delta_{ji}^p \left(\sum_{h \in K_{O_k}^o} s_{ij}^{hk} + \sum_{h \in K_{D_k}^d} t_{ij}^{hk} + \sum_{h \in K_{O_k}^d} r_{ij}^{hk} + \sum_{h \in K_{D_k}^o} u_{ij}^{hk} \right) \right\} z_p^k \Big] \\
&\quad - \sum_{k \in K} \sum_{p \in P^k} z_p^k \pi^k \quad (3.56)
\end{aligned}$$

以上のことから, $LF(z)$ は次のように整理できる .

$$LF(z) = \sum_{k \in K} \sum_{p \in P^k} \left\{ \sum_{(i,j) \in A} (\delta_{ij}^p \gamma_{ij}^k + \delta_{ji}^p \eta_{ji}^k) - \pi^k \right\} z_p^k \quad (3.57)$$

ここで,

$$\gamma_{ij}^k = c_{ij}^k + w_{ij}^k + \sum_{h \in K_{O^k}^o} s_{ij}^{kh} + \sum_{h \in K_{D^k}^d} t_{ij}^{kh} + \sum_{h \in K_{O^k}^d} u_{ij}^{kh} + \sum_{h \in K_{D^k}^o} v_{ij}^{hk} \quad \forall k \in K, (i, j) \in A \quad (3.58)$$

$$\eta_{ji}^k = c_{ji}^k + w_{ji}^k + \sum_{h \in K_{O^k}^o} s_{ij}^{hk} + \sum_{h \in K_{D^k}^d} t_{ij}^{hk} + \sum_{h \in K_{O^k}^d} r_{ij}^{kh} + \sum_{h \in K_{D^k}^o} r_{ij}^{hk} \quad \forall k \in K, (i, j) \in A \quad (3.59)$$

である.

以上のことから, 双対変数値が求められたときにパスフロー変数 z の被約費用は, (3.57) 式の係数

$$\sum_{(i,j) \in A} (\delta_{ij}^p \gamma_{ij}^k + \delta_{ji}^p \eta_{ij}^k) - \pi^k \quad \forall p \in P^k, k \in K \quad (3.60)$$

で与えられる.

$UPFSLR(\bar{P})$ が単体法などによって最適に解かれ, 最適な双対変数値が求められているものとする. これらの値を用いて価格付け問題を作成する. 被約費用が負であるパスフロー変数を見つけるには, 品種 $k \in K$ に対して, (3.60) 式の値を最小化するパス p を見つければ良い. したがって, $UPFSLR(\bar{P})$ における品種 k に関する価格付け問題は, 次のような問題 PRP^k に帰着される.

(PRP^k)

$$\text{最小化} \quad \sum_{p \in P^k} \left\{ \sum_{(i,j) \in A} (\delta_{ij}^p \gamma_{ij}^k + \delta_{ji}^p \eta_{ji}^k) - \pi^k \right\} z_p^k \quad (3.61)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad (3.62)$$

$$z_p^k \geq 0 \quad \forall p \in P^k \quad (3.63)$$

π^k は定数項であるので, PRP^k は, アーク (i, j) の長さが γ_{ij}^k , アーク (j, i) の長さが η_{ji}^k であるネットワーク上において, 始点・終点間の長さが最小となる各品種のパスを求める問題, すなわち最短路問題となる.

したがって, PRP^k は次のような品種 k の始点・終点間の最短路問題 SPP^k に帰着される.

(SPP^k)

$$\text{最小化} \quad \sum_{(i,j) \in A} (\gamma_{ij}^k \nu_{ij}^k + \eta_{ji}^k \nu_{ji}^k) \quad (3.64)$$

$$\text{条件} \quad \sum_{i \in N_n} \nu_{in}^k - \sum_{j \in N_n} \nu_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N \quad (3.65)$$

$$\nu_{ij}^k \geq 0, \nu_{ji}^k \geq 0 \quad \forall (i, j) \in A \quad (3.66)$$

品種 k に対する SPP^k を解き, k の始点・終点間の最短パス p^k と最短距離 μ^k を求める. $\mu^k - \pi^k < 0$ であれば, 被約費用が負であるパスフロー変数が見つかったことになる. アークの長さは非負であるので, SPP^k は Dijkstra 法 [25] によって解くことができる.

品種 $k (\in K)$ について, p^k を生成して \bar{P}^k に加え, 再度 $UPFSLR(\bar{P})$ を解き直していく. すべての品種に対して $\mu^k - \pi^k \geq 0$ となれば, すべての変数の被約費用が非負となり, $UPFSL$ が最適に解けたことになる.

パスフロー変数を生成した際に, まだ生成されていない強制制約式を生成し, 制約式として加える必要がある. 小規模な問題であっても, 生成される強制制約式の数は非常に多くなる. そこで, 生成されたパスに対して, 新たに $A_{\bar{P}^k}$ の要素となったアークについて, (3.38) および (3.39) 式を生成する. 一方, (3.40) ~ (3.43) 式に関しては, $UPFSLR(\bar{P})$ の最適解 \tilde{z} , \tilde{y} が満たさない次のような有効な制約式のみを問題に加える.

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h > \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^o, k \in K \quad (3.67)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h > \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{D^k}^d, k \in K \quad (3.68)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ij}^p z_p^h > \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.69)$$

$$\sum_{p \in \bar{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h > \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.70)$$

四つの式の左辺は, (3.48) 式の左辺と (3.49) 式の左辺の和によって求めることができる. $A_{\bar{P}^k}$ に含まれ, かつ四つの式の対象となる品種 k と h の組合せについて, \tilde{z} における (3.48) 式の左辺と (3.49) 式の左辺の値の和を計算し, これが \tilde{y} を超えれば, 有効な制約式と判断できる.

このような処理をしても強制制約式の数が非常に多くなる場合があり, 生成されるパスフロー変数も非常に多くなる場合がある. このため, 不必要なパスフロー変数および強制制約式を削除し, 問題の規模を逐次縮小する必要がある. そこで, パスフロー変数が現時点の $UPFSLR(\bar{P})$ の基底解に含まれていない場合は, 現時点で不必要な変数であると考えて, この変数とパスを削除し, $A_{\bar{P}^k}$ および $A_{\bar{P}^h}^{kh}$ を縮小する. また, 変数を削除した結果, 削除されたパスフロー変数を含まない強制制約式を削除する. さらに, スラック変数が 0 であるアクティブでない強制制約式 (3.50) ~ (3.54), すなわち

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h < \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^o, k \in K \quad (3.71)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h < \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{D^k}^d, k \in K \quad (3.72)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ij}^p z_p^h < \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.73)$$

$$\sum_{p \in \bar{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h < \tilde{y}_{ij} \quad \forall (i, j) \in A_{\bar{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.74)$$

である制約式は，現時点で不必要な制約であるため削除する．

UPFSL の解法

- [ステップ 1] 品種 $k (\in K)$ の初期パス集合を \bar{P}^k ，最終的な繰り返し回数を L ，問題の縮小を行う周期を l_{red} とする．品種 $k (\in K)$ について， $h \in K_{O^k}^o \cup K_{D^k}^o \cup K_{O^k}^d$ である品種対 (k, h) に対して $A_{\bar{P}^k}^{kh}$ を求める． $l := 1$ とする．
- [ステップ 2] $UPFSLR(\bar{P})$ を解き，双対変数 v, s, t, u, w を求め，最適解 \tilde{y} と \tilde{z} を求める．
- [ステップ 3] $l = l_{red}$ であれば，問題を縮小する．
- $UPFSLR(\bar{P})$ から基底変数でないパスフロー変数を削除し，対応するパスを $\bar{P}^k (k \in K)$ から削除する． $A_{\bar{P}^k}, A_{\bar{P}^k}^{kh}$ を更新する．
 - $UPFSLR(\bar{P})$ から (3.71) ~ (3.74) を満足する強制制約式を削除する．
- [ステップ 4] $UPFSLR(\bar{P})$ に (3.67) ~ (3.70) を満足する強制制約式を加える．
- [ステップ 5] すべての品種 $k (\in K)$ に対して，以下の操作を行う．
- SPP^k を解き，品種 k の始点・終点間の最短距離 μ^k と最短パス p^k を求める．
 - $\mu^k - \pi^k < 0$ であれば，パス p^k を \bar{P}^k に加え，パスフロー変数 $z_{p^k}^k$ を生成する．
 - パス p^k 上のアーク (i, j) が $A_{\bar{P}^k}$ が含まれていなければ，(3.38) および (3.39) 式を生成し， $UPFSLR(\bar{P})$ に加える．
 - $A_{\bar{P}^k}$ および $A_{\bar{P}^k}^{kh} (h \in K_{O^k}^o \cup K_{D^k}^o \cup K_{O^k}^d)$ を更新する．
- [ステップ 6] 追加されたパスがあれば， $l := l + 1$ とし，ステップ 2 へ戻る．そうでなければ， $L := l$ とし，終了する．

3.5.3 近似解法

$UPFSL$ または $UPFSLR(\bar{P})$ のデザイン変数の緩和解をもとにした三種類の近似解法と，変数を固定して問題を縮小する釘付けテストを示す．

限定した Minoux 法の改良法

$UPFSL$ の解法における繰り返し回数を l とし，そのときの $UPFSLR(\bar{P})$ の解を \tilde{y}^l とする．適当な閾値を $\epsilon_0 (0 < \epsilon_0 < 1)$ とし，アーク (i, j) に対して，次のように一部のデザイン変数を 0 か 1 に設定する．

$$y_{ij} = \begin{cases} 0 & \text{if } \tilde{y}_{ij}^l \leq \epsilon_0 \\ 1 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad (3.75)$$

(3.75) 式によって 1 とされたアークの集合を初期集合とした問題に対して，Minoux 法の改良法を適用する．繰り返しのたびに \tilde{y}^l が変化し，初期のアーク集合が多様化するため，多様な近似解を算出できることになり，これらの最良解を採用する．

ここでは、次のようなアークフロー変数を用いた定式化 $RFX(\tilde{y}^l)$ を用いる。パスフローではなく、アークフローによる定式化を用いるのは、最適解に含まれるすべてのパスが生成されていない可能性があるためである。

($RFY(\tilde{y}^l)$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.76)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (3.77)$$

$$x_{ij}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.78)$$

$$x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3.79)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (3.80)$$

$$y_{ij} = \begin{cases} 0 & \text{if } \tilde{y}_{ij}^l \leq \epsilon_0 \\ 1 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad (3.81)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (3.82)$$

ω 近傍局所探索法

アーク交換による局所探索法は、現在の解に含まれていないアークを付加し、現在の解に含まれているアークを削除したときの目的関数値の変化量を計算し、目的関数値が最も減少するようなアークの付加と削除を同時に行うアーク交換を行い、この操作を目的関数値が減少する間、繰り返す方法である。

l 回目の繰り返しにおいて、限定した Minoux 法の改良法により求めた現在までの最良の上界値と $\psi\%$ 以内の差の上界値が求めた場合、局所探索法により、Minoux 法の改良法により得られた解を初期解として解を改善する。アーク交換の組合せ数は非常に多いため、付加を検討するアークは、削除するアークの端点の近傍にある ω 個のノードを端点とするアークに限定し、これにより計算量を抑える。このような方法を ω 近傍局所探索法とよぶ。

釘付けテスト

0-1 変数に対する釘付けテストは、特定の 0-1 変数が 0 または 1 であるという制約条件を付けた緩和問題を解くことによって、0 または 1 に固定できる変数を見つけ、問題を縮小する方法である。一部の 0-1 変数が 0 または 1 であることが確定すれば、問題の規模が縮小し、かつ実行可能領域が狭まり、下界値が増加する。

UPFSL の解が得られた時点で、釘付けテストを実行する。まず、特定のデザイン変数 y_{ij} について、 $y_{ij} = 0$ という制約条件を付加した線形緩和問題を解き、下界値を求める。この下界値が現在までの最良の上界値を超えた場合、 $y_{ij} = 0$ であると上界値以下の解が存在しないことになり、 $y_{ij} = 1$ であることが確定する。同様に、 $y_{ij} = 1$ という制

約条件を付加した線形緩和問題を解き、下界値を求める。この下界値が上界値を超えた場合、 $y_{ij} = 1$ であると上界値以下の解が存在しないことになり、 $y_{ij} = 0$ であることが確定する。これらの操作を 0 または 1 に確定するアークがなくなるまで繰り返す。

釘付けテストにより縮小した問題を解き直し、下界値を再計算する。ただし、0-1 変数を 0 または 1 に固定する条件を付加した問題におけるパスフロー変数がすべて生成されているとは限らないので、固定された問題において列生成を行うことが必要となる。

限定した分枝限定法

\tilde{y}^l は繰り返しのたびに变化するため、 L 回目のときの最終的な解である \tilde{y}_{ij}^l が 0 に近くても、少し前の繰り返しではそうではない可能性がある。このような場合、デザイン変数を 0 に固定してしまうと $UPFS$ の最適解を排除する可能性がある。そこで、 $UPFSR(\tilde{P})$ の最適解である \tilde{y}^l ではなく、過去のデザイン変数の情報も考慮して、次のように \bar{y}^l を定める。

$$\bar{y}^l := \begin{cases} \tilde{y}^l & l = 1 \\ \rho \tilde{y}^l + (1 - \rho) \bar{y}^{l-1} & \forall l = 2, \dots, L \end{cases} \quad (3.83)$$

ここで、 ρ は $0 < \rho < 1$ である平滑化パラメータである。 \bar{y}^l は \tilde{y}^l を指数平滑化したものであり、最近の繰り返しにおけるデザイン変数値の情報の重みを大きく、古い情報の重みは小さくして算出したものである。

適当な閾値を $\epsilon_1, \epsilon_2 (0 \leq \epsilon_1 < \epsilon_2 < 1)$ とし、アーク (i, j) に対して、 L 回目の \bar{y}^L を用いて、次のように一部のデザイン変数を 0 か 1 に固定する。

$$y_{ij} = \begin{cases} 0 & \text{if } \bar{y}_{ij}^L \leq \epsilon_1 \\ 1 & \text{if } \bar{y}_{ij}^L \geq \epsilon_2 \end{cases} \quad \forall (i, j) \in A \quad (3.84)$$

(3.84) 式によって一部を 0 または 1 に固定して問題の規模を縮小し、かつ列生成法と行生成法によって生成された変数と制約式を用いた次のような問題 $PFX(\bar{y}^L, \tilde{P})$ を作成する。

$(PFX(\bar{y}^L, \tilde{P}))$

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in \tilde{P}^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (3.85)$$

$$\text{条件} \quad \sum_{p \in \tilde{P}^k} z_p^k = 1 \quad \forall k \in K \quad (3.86)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}, k \in K \quad (3.87)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ji}^p z_p^k \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}, k \in K \quad (3.88)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \tilde{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}^{kh}, h \in K_{O^k}^o, k \in K \quad (3.89)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \tilde{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}^{kh}, h \in K_{D^k}^d, k \in K \quad (3.90)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \tilde{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.91)$$

$$\sum_{p \in \tilde{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \tilde{P}^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall (i,j) \in A_{\tilde{P}^k}^{kh}, h \in K_{O^k}^d, k \in K \quad (3.92)$$

$$z_p^k \geq 0 \quad \forall k \in K, p \in \tilde{P}^k \quad (3.93)$$

$$y_{ij} = \begin{cases} 0 & \text{if } \bar{y}_{ij}^L \leq \epsilon_1 \text{ or } (i,j) \in A_0 \\ 1 & \text{if } \bar{y}_{ij}^L \geq \epsilon_2 \text{ or } (i,j) \in A_1 \end{cases} \quad \forall (i,j) \in A \quad (3.94)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \quad (3.95)$$

ここで, A_0 は釘付けテストにより $y_{ij} = 0$ が確定したアーク集合, A_1 は $y_{ij} = 1$ が確定したアーク集合である. また, \tilde{P}^k は $UPFSLR(\tilde{P})$ の最終的な繰り返しにおけるパス集合 \tilde{P}^k , \tilde{P} は最終的なパス集合 \tilde{P} である.

分枝限定法によって, $PFX(\bar{y}^L, \tilde{P})$ を解く. 得られる解は $UPFS$ の近似解となる. 0 または 1 に固定されていないデザイン変数が比較的少数であれば, 数理最適化ソフトウェアを用いて解くことができる. 0-1 変数が多くなる場合には, 計算時間の上限を設けて, $PFX(\bar{y}^L, \tilde{P})$ を解き, 最良解を求める. $PFX(\bar{y}^L, \tilde{P})$ の最良のデザイン変数解を \hat{y} とする. デザイン変数を \hat{y} に固定したフロー問題を解き, アークフロー \hat{x} および上界値を求める.

近似解法を含めた線形緩和法

近似解法を含めた線形緩和法の全体の流れを示す.

線形緩和法

[ステップ 1] 品種 $k(\in K)$ の初期パス集合を \tilde{P}^k , 最終的な繰り返し回数を L とする. 近似解法と問題の縮小を行う周期を l_{red} とし, $UB := \infty$ とする. 品種 $k(\in K)$ について, $h \in K_{O^k}^o \cup K_{D^k}^o \cup K_{O^k}^d$ である品種対 (k, h) に対して $A_{\tilde{P}^k}^{kh}$ を求める.

- $l := 1$ とする .
- [ステップ 2] $UPFSLR(\bar{P})$ を解き , 双対変数 v, s, t, u, w を求め , 最適解 \tilde{y} と \tilde{z} を求める . \tilde{y} を求める . $l \bmod l_{red} \neq 0$ であれば , ステップ 6 へ .
- [ステップ 3] 限定した Minoux 法の改良法および ω 近傍局所探索法を用いて $RFY(\tilde{y}^l)$ を解き , 上界値 UB_l を求める . $UB_l < UB$ であれば , $UB := UB_l$ とする .
- [ステップ 4] 基底に含まれないパスフロー変数とアクティブでない強制制約式を削除する .
- [ステップ 5] アクティブな強制制約式を追加する .
- [ステップ 6] 列生成と行生成を行い , パス , パスフロー変数 , 制約式を追加する . \bar{P} を更新する .
- [ステップ 7] 追加されたパスフロー変数があれば , $l := l + 1$ とし , ステップ 2 へ戻る . そうでなければ , $L := l, \tilde{P} := \bar{P}$ とする .
- [ステップ 8] 釘付けテストと , それに伴う列生成を行う . 問題を縮小する . 問題を解き , 下界値 LB を求める .
- [ステップ 9] 分枝限定法を用いて , $PFX(\tilde{y}^L, \tilde{P})$ を解き , 解 \hat{y}, \hat{x} を求める .
- [ステップ 10] \hat{y} を固定したフロー問題を解き , 上界値 UB_L を求める . $UB_L < UB$ であれば , $UB := UB_L$ とする .

3.6 数値実験

数値実験によって , 従来の解法と本章で示した近似解法の比較を行う . 数値実験には , アーク上のフロー費用とデザイン費用が比例するようなユークリッド平面上の問題例を用いる . なお , 使用したデータは www.rku.ac.jp/~katayama/networkdata に掲載している .

主な条件は次の通りである .

- 1) ノード : 1000×1000 のユークリッド平面上にランダムに発生させる .
- 2) アーク : 2 ノード間の内 , ノード間のユークリッド距離の短いものをアーク候補とする . ただし , 2 ノード間の距離を重みとした最小木に含まれるアークを含む .
- 3) 品種 : 品種に対応する始点・終点の対をランダムに選定する . ただし , 各ノードを始点または終点とする品種が少なくとも 1 品種は存在するものとする .
- 4) 問題数 : 同一ノード数について , 各 5 問とする .
- 5) ノード数 : Minoux の改良法では $10 \sim 1000$, 線形緩和法では $10 \sim 100$ とする . これらの問題を $N10$ 問題 , \dots , $N1000$ 問題とする .
- 6) アーク数 : ノード数の 5 倍とする .
- 7) 品種数 : ノード数の 5 倍とする .
- 8) デザイン費用 : ノード間のユークリッド距離に比例する .
- 9) フロー費用 : ノード間のユークリッド距離に比例する .

- 10) デザイン費用/フロー費用の比：10, 20, 30 とする．これらの問題を C10 問題，C20 問題，C30 問題とする．
- 11) 線形緩和法のパラメータ
- 平滑化係数 ρ : 0.2 とする．
 - 閾値 $\epsilon_0, \epsilon_1, \epsilon_2$: $\epsilon_0 = 0.2, \epsilon_1 = 0.1, \epsilon_2 = 0.85$ とする．
 - ω 局所探索法のパラメータ ω, ψ : $\omega = 10, \psi = 0.2\%$ とする．
 - 近似解法を実施する周期と問題の縮小の周期 l_{red} : 25 とする．
 - 計算時間：列生成と行生成の計算時間の上限を 10 時間，分枝限定法の計算時間の上限を 30 分とし，これらの時間に達した場合は計算を打ち切り，その時点での最良値を採用する．また，従来の解法であるバックワード法の計算時間の上限を 10 時間とする．
- 12) 使用計算機：CPU Pentium Core2-Duo 3.3GHz, メモリ 4GB, OS Windows XP
- 13) 使用言語：Minoux の改良法および従来の解法 Intel FORTRAN Ver9.1
線形緩和法 Visual Studio 2005 VB.NET
- 14) 最適化ソフトウェア：CPLEX Ver11.2
- 15) 上界値の誤差：ノード数 10 ~ 100
(各解法による上界値 - 下界値)/下界値 $\times 100$ の平均値
とする．下界値は Lagrange 緩和法および CPLEX によって得られた内の最良値とする．CPLEX の計算時間の上限を 10 時間とし，上限を超えた場合はその時点での最良解を採用する．なお，C10 問題と C20 問題はノード数 40 まで，C30 問題はノード数 30 までは数理最適化ソフトウェア CPLEX により求めた最適値を用いる．
- 16) 上界値の比較：ノード数 100 以上の上界値の比較には，次の値を用いる．
各解法による上界値/最良の上界値 $\times 100$ の平均値
最良の上界値は，比較を行う解法によって得られた内の最良値である．

問題の属性を表 3.1 および表 3.2 に示す．表 3.1 は，各問題におけるアーク上のデザイン費用とフロー費用の比であり，数字が大きいほどデザイン費用が大きい．表 3.2 は，各問題におけるノード数，アーク数と品種数である．

3.6.1 Minoux の改良法

本章で提案した二種類の Minoux の改良法と，従来の解法であるバックワードおよび Minoux 法との比較を行う．ノード数 10 ~ 100 までの問題に対する上界値の誤差を表 3.3 に示す．

バックワード法の誤差は，デザイン費用が安い C10 問題では 0.36 ~ 2.81% で平均 1.31%，C20 問題では 0.26 ~ 3.92% で平均 1.97%，デザイン費用が高い C30 問題では 0.10 ~ 3.66% で平均 2.14% である．Minoux 法の誤差は，C10 問題では 0.99 ~ 5.17% で

表 3.1 問題の属性 1

問題	デザイン費用・ フロー費用比
C10	10
C20	20
C30	30

表 3.2 問題の属性 2

問題	ノード	アーク	品種
N10	10	45	45
N20	20	100	100
N30	30	150	150
N40	40	200	200
N50	50	250	250
N60	60	300	300
N70	70	350	350
N80	80	400	400
N90	90	450	450
N100	100	500	500
N200	200	1000	1000
N300	300	1500	1500
N400	400	2000	2000
N500	500	2500	2500
N600	600	3000	3000
N700	700	3500	3500
N800	800	4000	4000
N900	900	4500	4500
N1000	1000	5000	5000

平均 3.31% , C20 問題では 1.03~7.33% で平均 4.62% , C30 問題では 0.66~7.78% で平均 5.07% である . 改良法 1 の誤差は , C10 問題では 0.36~3.22% で平均 1.56% , C20 問題では 0.26~4.17% で平均 2.07% , C30 問題では 0.10~3.89% で平均 2.30% である . 改良法 2 の誤差は , C10 問題では 0.36~3.02% で平均 1.46% , C20 問題では 0.26~3.98% で平均 2.00% , C30 問題では 0.10~3.71% で平均 2.24% である . どの解法においても , デザイン費用が高くなる , ノード数が多くなると誤差が大き . これは問題が難しくなるとともに , 下界値の精度が悪化することの影響もある .

解法の中では , バックワード法が大半の問題で誤差が最も小さい . 改良法 1 の誤差は , バックワード法と平均で 0.10~0.25% の差 , 最大で 0.57% の差がある . 改良法 2 の誤差は , バックワード法と平均で 0.03~0.15% の差 , 最大で 0.34% の差があるが , いくつかの問題ではバックワード法よりも良い解を算出している . 一方 , Minoux 法の誤差は , 平均で 3.31~5.07% と , 他の解法に比べて 2 倍程度の誤差である .

表 3.3 Minoux の改良法：上界値の誤差の比較

問題		バックワード (%)	Minoux (%)	改良法 1 (%)	改良法 2 (%)
C10	N10	0.36	0.99	0.36	0.36
	N20	0.38	1.72	0.50	0.41
	N30	0.37	2.12	0.37	0.37
	N40	0.55	3.25	0.74	0.79
	N50	0.74	2.83	1.08	0.81
	N60	0.99	3.71	1.22	1.23
	N70	1.93	4.05	2.20	2.08
	N80	2.59	4.83	3.16	2.77
	N90	2.81	5.17	3.22	3.02
	N100	2.39	4.46	2.69	2.73
	平均	1.31	3.31	1.56	1.46
C20	N10	0.26	1.03	0.26	0.26
	N20	1.47	3.65	1.45	1.47
	N30	0.41	3.08	0.38	0.38
	N40	0.56	3.26	0.89	0.79
	N50	1.67	3.95	1.62	1.66
	N60	2.56	5.86	2.68	2.43
	N70	2.89	5.36	2.94	2.87
	N80	3.92	6.90	3.94	3.98
	N90	3.78	7.33	4.17	3.81
	N100	2.20	5.75	2.39	2.35
	平均	1.97	4.62	2.07	2.00
C30	N10	0.10	0.66	0.10	0.10
	N20	1.20	2.98	1.17	1.17
	N30	0.69	2.62	0.73	0.75
	N40	1.54	4.45	1.74	1.72
	N50	2.03	4.75	2.30	2.27
	N60	2.64	6.47	2.84	2.62
	N70	2.62	6.35	2.82	2.79
	N80	3.66	7.78	3.89	3.67
	N90	3.59	7.69	3.82	3.71
	N100	3.32	6.92	3.60	3.62
	平均	2.14	5.07	2.30	2.24
平均	1.81	4.33	1.98	1.90	

ノード数 10~100 までの問題に対する計算時間を表 3.4 に示す。計算時間では、費用比の大小による影響ほとんど見られない。バックワード法の計算時間は 0.0~26.0 秒で平均 6.2~6.4 秒、Minoux 法で 0.0~0.1 秒で平均 0.0 秒、改良法 1 では 0.0~0.2 秒で平均 0.1 秒、改良法 2 では 0.0~0.2 秒で平均 0.1 秒である。Minoux 法が最も計算時間が短く、バックワード法はノード数とともに大幅に増加している。改良法 1 と 2 は、ノード数 100

表 3.4 Minoux の改良法：計算時間の比較

問題		バックワード (秒)	Minoux (秒)	改良法 1 (秒)	改良法 2 (秒)
C10	N10	0.0	0.0	0.0	0.0
	N20	0.0	0.0	0.0	0.0
	N30	0.2	0.0	0.0	0.0
	N40	0.6	0.0	0.0	0.0
	N50	1.5	0.0	0.1	0.1
	N60	3.0	0.0	0.1	0.1
	N70	5.8	0.0	0.1	0.1
	N80	9.9	0.1	0.1	0.1
	N90	16.1	0.1	0.2	0.2
	N100	24.9	0.1	0.2	0.2
	平均	6.4	0.0	0.1	0.1
C20	N10	0.0	0.0	0.0	0.0
	N20	0.0	0.0	0.0	0.0
	N30	0.2	0.0	0.0	0.0
	N40	0.6	0.0	0.0	0.0
	N50	1.5	0.0	0.0	0.0
	N60	3.1	0.0	0.1	0.1
	N70	5.8	0.0	0.1	0.1
	N80	10.1	0.1	0.1	0.1
	N90	16.0	0.1	0.2	0.2
	N100	24.8	0.1	0.2	0.2
	平均	6.2	0.0	0.1	0.1
C30	N10	0.0	0.0	0.0	0.0
	N20	0.0	0.0	0.0	0.0
	N30	0.2	0.0	0.0	0.0
	N40	0.6	0.0	0.0	0.0
	N50	1.4	0.0	0.1	0.1
	N60	3.2	0.0	0.1	0.1
	N70	5.8	0.0	0.1	0.1
	N80	10.2	0.1	0.1	0.1
	N90	16.2	0.1	0.2	0.2
	N100	26.0	0.1	0.2	0.2
	平均	6.4	0.0	0.1	0.1
平均	6.3	0.0	0.1	0.1	

の問題でも 0.2 秒以内に解を算出している。

ノード数 100~1000 までの問題に対する上界値の比較を表 3.5 に、計算時間を表 3.6 に示す。バックワード法では、ノード数 500 までは最良の解のほとんどを算出しているが、ノード数 600 以上の問題では計算時間が 10 時間に達したため計算を打ち切り、最終的な解を求めることができていない。Minoux 法は、C10 問題では最良値と 1.74~2.90% で平

表 3.5 Minoux の改良法：大規模問題：上界値の比較

問題		バックワード (%)	Minoux (%)	改良法 1 (%)	改良法 2 (%)
C10	N100	100.00	102.06	100.29	100.33
	N200	100.00	102.90	100.78	100.21
	N300	100.00	102.22	100.57	100.10
	N400	100.00	102.38	100.57	100.23
	N500	100.00	101.86	100.50	100.11
	N600	<i>t</i>	101.91	100.44	100.00
	N700	<i>t</i>	101.96	100.52	100.00
	N800	<i>t</i>	101.82	100.45	100.00
	N900	<i>t</i>	101.74	100.39	100.00
	N1000	<i>t</i>	101.74	100.41	100.00
	平均	—	102.06	100.49	100.10
C20	N100	100.06	103.59	100.24	100.20
	N200	100.00	103.55	100.67	100.19
	N300	100.00	103.06	100.53	100.15
	N400	100.00	103.54	100.63	100.16
	N500	100.00	103.01	100.76	100.18
	N600	<i>t</i>	102.93	100.37	100.00
	N700	<i>t</i>	103.08	100.55	100.00
	N800	<i>t</i>	103.26	100.57	100.00
	N900	<i>t</i>	102.94	100.50	100.00
	N1000	<i>t</i>	102.81	100.46	100.00
	平均	—	103.18	100.53	100.09
C30	N100 100	100.01	103.59	100.29	100.31
	N200 200	100.05	104.42	100.50	100.23
	N300 300	100.03	103.81	100.61	100.12
	N400 400	100.00	104.27	100.54	100.26
	N500 500	100.00	104.12	100.97	100.32
	N600 600	<i>t</i>	103.81	100.38	100.00
	N700 700	<i>t</i>	103.91	100.65	100.00
	N800 800	<i>t</i>	104.15	100.59	100.00
	N900 900	<i>t</i>	103.88	100.45	100.00
	N1000 1000	<i>t</i>	103.63	100.51	100.00
	平均	—	103.96	100.55	100.12
平均	—	103.07	100.52	100.10	

t:計算時間の上限

均 2.06% , C20 問題では 2.81 ~ 3.59% で平均 3.18% , C30 問題では 3.59 ~ 4.27% で平均 3.96% の差である . 改良法 1 は , C10 問題では最良値と 0.29 ~ 0.78% で平均 0.49% , C20 問題では 0.24 ~ 0.67% で平均 0.53% , C30 問題では 0.29 ~ 0.97% で平均 0.55% の差である . 改良法 2 は , C10 問題では最良値と 0.00 ~ 0.33% で平均 0.10% , C20 問題では 0.00

表 3.6 Minoux の改良法 : 大規模問題 : 計算時間の比較

問題		バックワード (秒)	Minoux (秒)	改良法 1 (秒)	改良法 2 (秒)
C10	N100	24.9	0.1	0.2	0.2
	N200	515.7	0.6	1.8	2.4
	N300	3087.4	1.9	6.6	9.2
	N400	10503.9	4.5	16.7	23.2
	N500	26853.6	9.0	33.9	47.2
	N600	<i>t</i>	15.2	60.6	88.1
	N700	<i>t</i>	24.0	98.7	148.7
	N800	<i>t</i>	35.2	148.2	226.9
	N900	<i>t</i>	50.1	216.6	333.6
	N1000	<i>t</i>	69.1	299.0	459.8
	平均	—	21.0	88.2	133.9
C20	N100	24.8	0.1	0.2	0.2
	N200	517.2	0.6	2.0	2.7
	N300	3163.5	1.9	7.6	10.1
	N400	10830.0	4.5	19.0	25.5
	N500	28186.8	9.1	38.8	51.3
	N600	<i>t</i>	15.4	69.3	97.4
	N700	<i>t</i>	24.4	113.3	162.3
	N800	<i>t</i>	35.7	172.0	250.1
	N900	<i>t</i>	51.0	253.5	371.6
	N1000	<i>t</i>	70.7	345.0	507.2
	平均	—	21.3	102.1	147.8
C30	N100	26.0	0.1	0.2	0.2
	N200	540.6	0.6	2.1	2.8
	N300	3220.6	1.9	8.1	10.5
	N400	10728.4	4.6	20.6	26.7
	N500	27952.5	9.2	41.9	54.6
	N600	<i>t</i>	15.6	75.8	101.8
	N700	<i>t</i>	24.7	122.5	170.6
	N800	<i>t</i>	36.4	187.9	263.8
	N900	<i>t</i>	51.7	276.9	391.6
	N1000	<i>t</i>	71.2	378.2	537.1
	平均	—	20.6	106.0	148.2
平均	—	21.0	98.8	143.3	

t: 計算時間の上限

~0.20% で平均 0.09%, C30 問題では 0.00~0.31% で平均 0.12% の差である。Minoux 法は他の解法よりも平均で 2.06~3.96% 悪く、デザイン費用が高くなると差が大きい。また、ノード数 600 以上では改良法 2 が最良の解を算出している。

計算時間では、デザイン費用の大小による影響は見られない。バックワード法の計算時

表 3.7 線形緩和法：上界値の誤差の比較

問題		バックワード (%)	Minoux (%)	CPLEX (%)	線形緩和 (%)
C10	N10	0.36	0.99	0.00	0.00
	N20	0.38	1.72	0.00	0.00
	N30	0.37	2.12	0.00	0.00
	N40	0.55	3.25	0.00	0.00
	N50	0.74	2.83	0.18	0.29
	N60	0.99	3.71	0.48	0.61
	N70	1.93	4.05	1.15	1.36
	N80	2.59	4.83	2.48	2.28
	N90	2.81	5.17	3.01	2.25
	N100	2.39	4.46	2.54	1.91
	平均	1.31	3.31	0.98	0.87
C20	N10	0.26	1.03	0.00	0.00
	N20	1.47	3.65	0.00	0.00
	N30	0.41	3.08	0.00	0.01
	N40	0.56	3.26	0.00	0.01
	N50	1.67	3.95	0.91	1.00
	N60	2.56	5.86	1.70	1.68
	N70	2.89	5.36	2.66	1.96
	N80	3.92	6.90	5.36	3.00
	N90	3.78	7.33	5.55	3.26
	N100	2.20	5.75	5.59	1.87
	平均	1.97	4.62	2.18	1.28
C30	N10	0.10	0.66	0.00	0.00
	N20	1.20	2.98	0.00	0.00
	N30	0.69	2.62	0.00	0.00
	N40	1.54	4.45	0.73	0.73
	N50	2.03	4.75	1.66	1.40
	N60	2.64	6.47	2.39	1.87
	N70	2.62	6.35	2.66	2.15
	N80	3.66	7.78	23.55	2.71
	N90	3.59	7.69	8.43	3.29
	N100	3.32	6.92	13.82	3.05
	平均	2.14	5.07	5.32	1.52
平均	1.81	4.33	2.83	1.22	

間はノード数 500 で 8 時間程度となり、膨大な計算時間が必要となる。Minoux 法は平均で 20.6~21.0 秒であり、ノード数が 1000 でも 72 秒以内で解を算出している。改良法 1 は平均で 88.2~106.0 秒であり、ノード数の増加とともに Minoux 法との計算時間の差が開き、ノード数 1000 では 299.0~378.2 秒を必要としている。また、改良法 2 は平均で 133.9~148.2 秒であり、ノード数の増加とともに Minoux 法との差が開き、ノード数

表 3.8 線形緩和法：計算時間の比較

問題		バックワード (秒)	Minoux (秒)	CPLEX (秒)	線形緩和 (秒)
C10	N10	0.0	0.0	0.3	4.0
	N20	0.0	0.0	22.4	67.4
	N30	0.2	0.0	456.4	429.6
	N40	0.6	0.0	5156.4	1553.2
	N50	1.5	0.0	19345.1	4197.0
	N60	3.0	0.0	28998.8	8333.4
	N70	5.8	0.0	<i>t</i>	18094.2
	N80	9.9	0.1	<i>t</i>	30202.6
	N90	16.1	0.1	<i>t</i>	<i>t</i>
	N100	24.9	0.1	<i>t</i>	<i>t</i>
	平均	6.2	0.0	19797.9	13851.3
C20	N10	0.0	0.0	1.2	11.0
	N20	0.0	0.0	102.3	290.2
	N30	0.2	0.0	2620.8	2220.2
	N40	0.6	0.0	14110.0	7573.6
	N50	1.5	0.0	32319.9	18532.8
	N60	3.1	0.0	30861.8	<i>t</i>
	N70	5.8	0.0	<i>t</i>	<i>t</i>
	N80	10.1	0.1	<i>t</i>	<i>t</i>
	N90	16.0	0.1	<i>t</i>	<i>t</i>
	N100	24.8	0.1	<i>t</i>	<i>t</i>
	平均	6.2	0.0	22401.6	21906.1
C30	N10	0.0	0.0	0.7	15.6
	N20	0.0	0.0	115.5	1010.4
	N30	0.2	0.0	2313.0	5314.4
	N40	0.6	0.0	30648.7	16889.8
	N50	1.4	0.0	<i>t</i>	<i>t</i>
	N60	3.2	0.0	<i>t</i>	<i>t</i>
	N70	5.8	0.0	<i>t</i>	<i>t</i>
	N80	10.2	0.1	<i>t</i>	<i>t</i>
	N90	16.2	0.1	<i>t</i>	<i>t</i>
	N100	26.0	0.1	<i>t</i>	<i>t</i>
	平均	6.4	0.0	24907.8	24932.3
平均	6.3	0.0	22369.1	20229.9	

t:計算時間の上限

1000 では 459.8 ~ 537.1 秒を必要としている .

3.6.2 線形緩和法

本章で提案した線形緩和法と従来の解法の比較を行う。従来の解法であるバックワード法, Minoux 法, および数理最適化ソフトウェア CPLEX による分枝限定法による上界値と比較する。なお, CPLEX の計算時間の上限を 10 時間とし, その時点で求められた最良の上界値を用いる。従来の解法による上界値と線形緩和法による上界値の平均誤差を表 3.7 に, 平均計算時間を表 3.8 に示す。

バックワード法の誤差は, デザイン費用が安い C10 問題では 0.36~2.81% で平均 1.31%, C20 問題では 0.26~3.92% で平均 1.97%, デザイン費用が高い C30 問題では 0.10~3.66% で平均 2.14% である。Minoux 法の誤差は, C10 問題では 0.99~5.17% で平均 3.31%, C20 問題では 1.03~7.33% で平均 4.62%, C30 問題では 0.66~7.78% で平均 5.07% である。CPLEX は, C10 問題と C20 問題ではノード数 40 まで, C30 問題ではノード数 30 までの問題において最適解を求めることができた。CPLEX による誤差は, C10 問題では 0.00~3.01% で平均 0.98%, C20 問題では 0.00~5.59% で平均 2.18%, C30 問題では 0.00~23.55% で平均 5.32% である。線形緩和法では, C10 問題はノード数 40 まで, C20 問題はノード数 20 まで, C30 問題はノード数 30 までの問題において最適解を求めることができた。線形緩和法による誤差は, C10 問題では 0.00~2.28% で平均 0.87%, C20 問題では 0.00~3.26% で平均 1.28%, C30 問題では 0.00~3.29% で平均 1.52% である。すべての解法で, デザイン費用が高くなる, ノード数が多くなると誤差が大きいの。これは問題が難しくなるとともに, 下界値の精度が悪くなることも要因の一つである。

バックワード法は, 規模の小さな問題では CPLEX に劣っているが, ノード数 80~90 以上の問題では良い解を算出している。Minoux 法の誤差は, CPLEX と平均で 0.25~2.44% の差がある。CPLEX は, 規模の大きな問題では計算時間の上限に達しているため, 誤差が大きい。C30 問題のノード数 80 の問題の 1 問で誤差が 30% を超えたものがあるため, 平均誤差が 23.55% と極端に大きい。また, C30 問題のノード数 100 の問題では平均誤差が 13% を超えている。

線形緩和法の誤差は, バックワード法に対して平均で 0.44~0.69% 良く, すべての問題においてバックワード法より良い解が求められている。また, CPLEX に対しては, 平均で 0.09~3.80% 良い解が求められている。特に, C10 問題ではノード数 80 以上, C20 問題ではノード数 60 以上, C30 問題ではノード数 50 以上の問題に対して, 比較した解法の中で最も良い解を算出している。

バックワード法の計算時間は 0.0~26.0 秒で平均 6.2~6.4 秒, Minoux 法は 0.0~0.1 秒で平均 0.0 秒と短時間で解を求めることができています。CPLEX の計算時間は, C10 問題では 0.3~36000.0 秒で平均 19797.9 秒, C20 問題では 1.2~36000.0 秒で平均 22401.6 秒, C30 問題では 0.7~36000.0 秒で平均 24907.8 秒である。なお, C10 問題と C20 問題のノード数 70 以上, および C30 問題のノード数 50 以上の問題では計算時間の上限であ

る 10 時間に達したため計算を打ち切っている。一方、線形緩和法の計算時間は、C10 問題では 4.0 ~ 36000.0 秒で平均 13851.3 秒、C20 問題では 11.0 ~ 36000.0 秒で平均 21906.1 秒、C30 問題では 15.6 ~ 36000.0 秒で平均 24932.3 秒である。なお、C10 問題のノード数 90 以上、C20 問題のノード数 60 以上、および C30 問題のノード数 50 以上の問題では、計算時間が 10 時間に達している。線形緩和法は生成するパス数が非常に大きくなり、特にデザイン費用の高い問題ではその傾向が顕著となるため、ノード数の増加とともに計算時間が大幅に増加している。規模の小さな問題では線形緩和法よりも CPLEX の計算時間が短い、ノード数が 40 ~ 50 以上の規模の大きな問題では、線形緩和法の計算時間が相対的に短い。

3.7 本章のまとめ

本章では、容量制約のないネットワーク設計問題に対して、強い妥当不等式と二種類の近似解法を示した。提案した妥当不等式は、同一ノードを始点とする品種と終点とする品種のフローが同一のアーキ上で同じ向きに移動するフローが存在しないことを用いた式である。第一の解法は、Minoux 法の評価に厳密性を加える改良を行った解法である。第二の解法は、妥当不等式も含むパスフローを定式化に対する列生成法と行生成法を用いた線形緩和法である。最後に、数値実験を行い、提案した解法の有効性を示した。

本章では、容量制約のないネットワーク設計問題に対して、以下のような成果を示した。

- 1) 強い妥当不等式である非逆流不等式を示した。

最適解において、同一ノードを始点とする品種と終点とする品種のフローが同一のアーキ上で同じ向きに移動するフローが存在しないことを用いた新たな二種類の強い妥当不等式である非順流不等式を示した。

- 2) 非逆流不等式と非順流不等式を制約式として含むアーキフローによる定式化とパスフローによる定式化を示した。

強い妥当不等式を含むため、これらの定式化は強い定式化となり、緩和問題から求められる下界値が良いものとなる。しかし、非常に多くの制約を含むため、直接的に解くことは困難な問題となる。

- 3) Minoux 法をもとにした二種類の改良法を提案した。

第一の改良法は、Minoux 法におけるアーキ削除時の目的関数の評価に厳密性を加える改良を行った解法である。第二の改良法は、第一の方法に加え、アーキ削除時の目的関数の初期評価に厳密性を加える改良を行った解法である。

- 4) 強い妥当不等式を含むパスフロー変数を用いた定式化に対して、列生成法と行生成法を用いた線形緩和法を示した。

この定式化の線形緩和問題に対して、限定主問題、被約費用および価格付け問題を示し、価格付け問題が最短路問題に帰着されることを示した。線形緩和問題に対して、変数を追加する列生成法と制約式を追加する行生成法を示した。また、不要

な変数と制約式を削除する問題の縮小方法を示した。さらに、釘付けテストを示し、線形緩和解をもとにした限定した Minoux 法の改良法、局所探索法、および分枝限定法を用いた近似解法を示した。

- 5) 数値実験により、提案する二つの解法と従来の近似解法の比較を行い、提案した解法の有効性を示した。

Minoux 法の改良法に対して、従来の解法であるバックワード法、Minoux 法と比較した数値実験を行った。線形緩和法に対して、バックワード法、Minoux 法、および CPLEX の分枝限定法と比較した数値実験を行い、提案した解法の有効性を示した。

- 6) 問題の属性と適用すべき解法

短時間で解を求めたい場合、ノード数 200、アーク数 1000 程度までの規模の問題に対してはバックワード法を適用し、ノード数 1000、アーク数 5000 程度までの規模の問題に対しては提案した Minoux の改良法を適用し、それ以上の規模の問題に対しては Minoux 法を適用することが良いと考える。

精度の高い解を求めたい場合、ノード数 50、アーク数 250 程度までの規模の問題に対しては CPLEX を利用し、ノード数 100、アーク数 500 程度までの規模の問題に対しては提案した線形緩和法を適用し、ノード数 500、アーク数 2500 程度までの規模の問題に対してはバックワード法を適用し、それ以上の規模の問題に対しては提案した Minoux の改良法を適用することが良いと考える。

今後の課題としては、次の項目が挙げられる。

- 1) 新たな妥当不等式の提示

本章では、新たな非順流不等式を提案した。しかし、規模の大きな問題では非順流不等式を満たす非 0-1 の緩和解が数多く生成され、その値も 0 または 1 から離れたものも多くなる。そのため、より強力な妥当不等式を提示することが課題である。

- 2) Minoux 法の改良法を用いた拡張モデルの開発

この解法は、大規模な問題に高速に適用できる解法である。そのため、タブー探索法などの繰り返し型のメタ解法を組み込むことが可能である。また、アーク削除に制限を加えることによって、輸送距離や輸送時間の制限、ノード経由回数の制限などの実際的な付帯制約をもつ問題に拡張できる可能性がある。

- 3) 線形緩和法の計算時間の短縮と拡張モデルの拡張

容量制約のない問題では、非常に多くのパスと制約式が生成され、多くの計算時間が必要となる。このため、さらなる不要なパスの削除、不要なパスを生成しないなどの問題規模の縮小化による計算時間の短縮が必要である。また、分枝限定法に移行する際の自由変数の数の適切な制御方法や、分枝限定法への列生成法と行生成法の組み込みが必要である。また、パスを変数として扱っていることから、生成するパスを限定することによって、輸送距離や輸送時間の制限、ノード経由回数の制

限などの実際的な付帯制約をもつ問題に拡張できる可能性がある。

第 4 章

容量制約のないネットワーク設計問題の Lagrange 緩和法

4.1 本章の概要

3 章で示した容量制約のないネットワーク設計問題 (UND) に対する近似解法によって得られた近似解の良さを保証するためには、目的関数の下界値が必要になる。近似解の良さをより正確に表すためには、より良い下界値を求めることが必要である。また、上界値により近い下界値をもつ解からは、良い近似解を導出できる可能性がある。そのため、 UND に対して、従来の下界値を求める解法よりも計算時間は大きくなっても良いが、より良い下界値を求める解法が必要となる。

本章では、3 章と同じ UND を対象として、 UND に対する三種類の Lagrange 緩和法を提案する。Lagrange 緩和法は、Lagrange 緩和問題とその解法、Lagrange 乗数の更新、Lagrange ヒューリスティックを組み合わせた解法であり、組合せ最適化問題に適用されて成果を挙げている手法である。

第一の解法は、Holmberg–Hellstrand[42] が示したアークが向きをもつ設計問題を対象とした Lagrange 緩和法を向きをもたない問題に適用した方法である。第二の解法は、始点を同じにする品種に対する非逆流不等式を制約式に含む定式化に対する Lagrange 緩和法である。第三の解法は、非逆流不等式と非順流不等式を含む定式化に対する Lagrange 緩和法である。

UND の前提条件と記号の定義を示した後、アークフローを用いた定式化を示す。続いて、三つの Lagrange 緩和問題とその解法を示す。さらに、提案する三種類の Lagrange 緩和法の有効性を示すための数値実験の結果を示す。最後に、本章のまとめを記述する。

4.2 問題の定式化

UND の前提条件と記号の定義を示した後、アークフローを用いた三種類の定式化を示す。

4.2.1 前提条件と記号の定義

前提条件

- ノード集合が与えられている .
- アーク集合が与えられている .
- アークは向きをもたない .
- アークの処理量には , 容量である上限を考慮しない .
- アークには , 非負のデザイン費用が与えられている .
- アークには , 単位当たりの非負のフロー費用が与えられている .
- 複数の品種からなる品種集合が与えられている .
- 各品種の需要は 1 である .
- 各品種の需要は , 始点から終点までのパス上を移動する .
- ネットワークは連結する .

品種の始点や終点でないノードが存在する場合 , 各品種の始点・終点間は連結するという条件に置き換える .

記号の定義

- N : ノード集合
- A : アーク集合
- K : 品種集合
- N_n : ノード n に接続するアークの他方の端点の集合
- P^k : 品種 k の取り得るパス集合
- K_n^o : ノード n を始点とする品種の集合
- K_n^d : ノード n を終点とする品種の集合
- O^k : 品種 k の始点
- D^k : 品種 k の終点
- $K_{O^k}^o$: 品種 k の始点 O^k を始点とする品種の集合
- $K_{D^k}^d$: 品種 k の終点 D^k を終点とする品種の集合
- $K_{O^k}^d$: 品種 k の始点 O^k を終点とする品種の集合
- c_{ij}^k : アーク (i, j) 上における品種 k の単位当たりの非負のフロー費用
- f_{ij} : アーク (i, j) の非負のデザイン費用
- x_{ij}^k : 品種 k のフローがアーク (i, j) 上を $i \rightarrow j$ 向きに移動する量を表すアークフロー変数 ; 非負の連続変数
- y_{ij} : アーク (i, j) を選択するとき 1 , そうでないとき 0 であるデザイン変数 ; 0-1 変数
- Y : ネットワークが連結するようなデザイン変数の集合

- v_n^k : 品種 k , ノード n に関するフロー保存式に対する双対変数 ; 連続変数
- t_{ij}^{kh} : アーク (i, j) における終点を同一にする品種 k と h に関する非逆流不等式に対する双対変数 ; 非負の連続変数
- u_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $i \rightarrow j$ 向き非順流不等式に対する双対変数 ; 非負の連続変数
- r_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $j \rightarrow i$ 向き非順流不等式に対する双対変数 ; 非負の連続変数
- s_n^k : 品種 k , ノード n に関するフロー保存式に対する劣勾配 ; 整数変数
- p_{ij}^{kh} : アーク (i, j) における終点を同一にする品種 k と h に関する非逆流不等式に対する劣勾配 ; 整数変数
- q_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $i \rightarrow j$ 向き非順流不等式に対する劣勾配 ; 整数変数
- g_{ij}^{kh} : アーク (i, j) における品種 k の始点と終点を同一にする品種 h に関する $j \rightarrow i$ 向き非順流不等式に対する劣勾配 ; 整数変数

問題の定義

ノード集合 N , デザイン費用 f とフロー費用 c をもつ向きをもたないアーク集合 A , 品種の需要 1 をもつ品種集合 K が与えられている. このとき, フロー費用とデザイン費用の合計を最小にするアーク集合 $A' (\subseteq A)$ とフロー x または z を求めよ.

4.2.2 定式化

UND のアークフローによる基本的な定式化を $UAFB$ とする. $UAFB$ は, 次のようになる.

($UAFB$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (4.1)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (4.2)$$

$$x_{ij}^k + x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (4.3)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4.4)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.5)$$

(4.1) 式は目的関数であり, フロー費用とデザイン費用の総和を最小化する. (4.2) 式はフロー保存式であり, ノードに流入するフローと流出するフローの差が, 品種 k の始点であれば -1 , 終点であれば 1 , その他のノードであれば 0 であることを表す. この式は, 各品種について, 必ず始点から終点まで需要が移動することを保証する. (4.3) 式は, 品種 k

におけるフロー変数とデザイン変数の関係を表す強制制約式である．アーク (i, j) が選択されるときにはこのアーク上を移動する品種 k のフローは最大で 1 となり，アーク (i, j) が選択されないときにはこのアーク上の品種 k のフローは 0 となることを表す．(4.4) 式はアークフロー変数の非負制約であり，(4.5) 式はデザイン変数の 0-1 条件である．

(4.3) 式の代わりに始点と同じである品種に対する非逆流不等式を用い，ネットワークの連結条件を制約式として加えた定式化を $UAFS$ とする．この定式化は， $UAFB$ よりも強い定式化となる．

($UAFS$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (4.6)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (4.7)$$

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (4.8)$$

$$\mathbf{y} \in Y \quad (4.9)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4.10)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.11)$$

(4.8) 式は，非逆流不等式である．左辺は品種 k の $i \rightarrow j$ 向きのフローと k と始点と同じである品種 h の $j \rightarrow i$ 向きのフローの和であり，これがアーク (i, j) が選択されるときに最大で 1，選択されないときに 0 となることを表す．(4.9) 式は， $y_{ij} = 1$ からなるアークで構成されるネットワークが連結することを表す．

$UAFS$ に，終点と同じである品種に対する非逆流不等式と二種類の非順流不等式を制約式として加えた拡張した定式化を $UAFE$ とする．この定式化は， $UAFS$ よりも強い定式化となる．

($UAFE$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (4.12)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (4.13)$$

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K, (i, j) \in A \quad (4.14)$$

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (4.15)$$

$$x_{ij}^k + x_{ij}^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.16)$$

$$x_{ji}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.17)$$

$$\mathbf{y} \in Y \quad (4.18)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4.19)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.20)$$

(4.15) 式は, 左辺が品種 k の $i \rightarrow j$ 向きのフローと k と終点と同じである品種 h の $j \rightarrow i$ 向きのフローの和であり, これがアーク (i, j) が選択されるときに最大で 1, 選択されないときに 0 となることを表す非逆流不等式である. (4.16) 式は, 左辺が品種 k の $i \rightarrow j$ 向きのフローと k の始点が終点である品種 h の $i \rightarrow j$ 向きのフローの和であり, これがアーク (i, j) が存在するときに最大で 1, 存在しないときに 0 となることを表す非順流不等式である. (4.17) 式は, 左辺が品種 k の $j \rightarrow i$ 向きのフローと k の始点が終点である品種 h の $j \rightarrow i$ 向きのフローの和であり, これがアーク (i, j) が選択されるときに最大で 1, 選択されないときに 0 となることを表す非順流不等式である.

4.3 基本的な定式化に対する Lagrange 緩和法

この節の Lagrange 緩和法は, Holmberg–Hellstrand[42] が示したアークが向きをもつ設計問題を対象とした Lagrange 緩和法を向きをもたない問題である $UAFB$ に適用した方法である.

4.3.1 Lagrange 緩和問題とその解法

$UAFB$ に対して, Lagrange 乗数 v を用いてフロー保存式 (4.2) を Lagrange 緩和した問題 LGB を作成する. はじめに, LGB の目的関数値を ϕ とし, ϕ を整理する.

$$\begin{aligned}
\phi &= \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \\
&+ \sum_{k \in K} v_{O^k}^k \left(-1 - \sum_{i \in N_{O^k}} x_{iO^k}^k + \sum_{j \in N_{O^k}} x_{O^k j}^k \right) + \sum_{k \in K} v_{D^k}^k \left(1 - \sum_{i \in N_{D^k}} x_{iD^k}^k + \sum_{j \in N_{D^k}} x_{D^k j}^k \right) \\
&\quad + \sum_{k \in K} \sum_{n \in N \setminus \{O^k, D^k\}} v_n^k \left(- \sum_{i \in N_n} x_{in}^k + \sum_{j \in N_n} x_{nj}^k \right) \\
&= \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} + \sum_{(i,j) \in A} f_{ij} y_{ij}
\end{aligned} \tag{4.21}$$

したがって, LGB は次のように表される.

(LGB)

$$\begin{aligned}
\text{最小化} \quad & \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} \\
& \quad + \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{4.22}
\end{aligned}$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \tag{4.23}$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \tag{4.24}$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \tag{4.25}$$

適当な v を与えると目的関数の第一項である $\sum_{k \in K} (v_{D^k}^k - v_{O^k}^k)$ は定数項として扱えるため, LGB はアーク (i, j) 毎の独立した次のような問題 LGB_{ij} に分割できる.

(LGB_{ij})

$$\text{最小化 } \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} + f_{ij} y_{ij} \quad (4.26)$$

$$\text{条件 } x_{ij}^k + x_{ji}^k \leq y_{ij} \quad \forall k \in K \quad (4.27)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K \quad (4.28)$$

$$y_{ij} \in \{0, 1\} \quad (4.29)$$

LGB_{ij} において, y_{ij} は 0 または 1 のどちらかである. そこで, $y_{ij} = 1$ と $y_{ij} = 0$ の場合に分けて考える.

$y_{ij} = 1$ である場合, $f_{ij} y_{ij}$ は定数項となるので, LGB_{ij} は次のような品種 k 毎の独立した問題 LGB_{ij}^{k1} に分割できる.

(LGB_{ij}^{k1})

$$\text{最小化 } (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \quad (4.30)$$

$$\text{条件 } x_{ij}^k + x_{ji}^k \leq 1 \quad (4.31)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad (4.32)$$

(4.31) 式は, 高々一方向のフローの存在を示す制約式となる. x_{ij}^k, x_{ji}^k の上限は 1 であり, かつ最小化問題である. そのため, 解が正であるためには, その係数が負である場合に限られる. したがって, LGB_{ij}^{k1} の最適値 $\tilde{\phi}^{k1}$ は $\min(0, c_{ij}^k - v_j^k + v_i^k)$ と $\min(0, c_{ji}^k - v_i^k + v_j^k)$ の小さい方となる.

$$\begin{aligned} \tilde{\phi}^{k1} &= \min\{\min(0, c_{ij}^k - v_j^k + v_i^k), \min(0, c_{ji}^k - v_i^k + v_j^k)\} \\ &= \min(0, c_{ij}^k - v_j^k + v_i^k, c_{ji}^k - v_i^k + v_j^k) \end{aligned} \quad (4.33)$$

LGB_{ij}^{k1} の最適解 \tilde{x} は, 次のようになる.

$$\tilde{x}_{ij}^k = \begin{cases} 1 & \text{if } c_{ij}^k - v_j^k + v_i^k < 0 \text{ and } c_{ij}^k - v_j^k + v_i^k \leq c_{ji}^k - v_i^k + v_j^k \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.34)$$

$$\tilde{x}_{ji}^k = \begin{cases} 1 & \text{if } c_{ji}^k - v_i^k + v_j^k < 0 \text{ and } c_{ji}^k - v_i^k + v_j^k < c_{ij}^k - v_j^k + v_i^k \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.35)$$

一方, $y_{ij} = 0$ である場合, LGB_{ij}^k の最適値は 0 である. したがって, LGB_{ij} はアークフロー変数 x を用いない次のような問題 $LGBY_{ij}$ に帰着できる.

(LGBY_{ij})

$$\text{最小化 } \left\{ \sum_{k \in K} \min(0, c_{ij}^k - v_j^k + v_i^k, c_{ji}^k - v_i^k + v_j^k) + f_{ij} \right\} y_{ij} \quad (4.36)$$

$$\text{条件 } y_{ij} \in \{0, 1\} \quad (4.37)$$

LGBY_{ij} の最適解は、係数が負であれば 1, そうでなければ 0 となり, 容易に解くことができる.

以上のことから, LGB はフロー変数 x を用いない次のような問題 LGBY に置き換えることができる.

(LGBY)

$$\text{最小化 } \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \left\{ \sum_{k \in K} \min(0, c_{ij}^k - v_j^k + v_i^k, c_{ji}^k - v_i^k + v_j^k) + f_{ij} \right\} y_{ij} \quad (4.38)$$

$$\text{条件 } y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.39)$$

LGBY および LGB の最適解 \hat{y} は,

$$\hat{y}_{ij} = \begin{cases} 1 & \text{if } \sum_{k \in K} \min(0, c_{ij}^k - v_j^k + v_i^k, c_{ji}^k - v_i^k + v_j^k) + f_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad (4.40)$$

となる. また, LGB の最適解 \hat{x} は次のようになる.

$$\hat{x}_{ij}^k = \begin{cases} \tilde{x}_{ij}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \quad \hat{x}_{ji}^k = \begin{cases} \tilde{x}_{ji}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.41)$$

\hat{y} を LGBY の目的関数に代入すれば, LGB の最適値, すなわち UND の下界値を求めることができる. LGB の最適値を $\hat{\phi}$ とすると,

$$\hat{\phi} = \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \left[\min \left\{ 0, \sum_{k \in K} \min(0, c_{ij}^k - v_j^k + v_i^k, c_{ji}^k - v_i^k + v_j^k) + f_{ij} \right\} \right] \quad (4.42)$$

4.3.2 劣勾配法

\hat{y} , \hat{x} が求められたときに, LGB において Lagrange 乗数 v を変数と見なした最大化問題 LDB を考える.

(LDB)

$$\begin{aligned} & \text{最大化 } \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k \hat{x}_{ij}^k + c_{ji}^k \hat{x}_{ji}^k) + \sum_{(i,j) \in A} f_{ij} \hat{y}_{ij} \\ & + \sum_{k \in K} v_{O^k}^k \left(-1 - \sum_{i \in N_{O^k}} \hat{x}_{iO^k}^k + \sum_{j \in N_{O^k}} \hat{x}_{O^k j}^k \right) + \sum_{k \in K} v_{D^k}^k \left(1 - \sum_{i \in N_{D^k}} \hat{x}_{iD^k}^k + \sum_{j \in N_{D^k}} \hat{x}_{D^k j}^k \right) \\ & + \sum_{k \in K} \sum_{n \in N \setminus \{O^k, D^k\}} v_n^k \left(- \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k \right) \quad (4.43) \end{aligned}$$

\hat{y}, \hat{x} は組合せ最適化問題である LGB の解であり, v が変化したときに離散的に変化するため, LDB の目的関数値も離散的に変化する. このため, LDB は微分不可能な目的関数をもつ最適化問題となり, 微分不可能な問題に対する解法が必要となる.

そこで, v に対して劣勾配法 [54] を適用する. LDB の目的関数は v の 1 次関数であることから, v に関する劣勾配 s を v の係数, すなわち,

$$s_n^k = \begin{cases} -1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = O^k \\ 1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = D^k \\ - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{otherwise} \end{cases} \quad \forall k \in K, n \in N \quad (4.44)$$

とする.

LDB の目的関数 (4.43) の値を改善する可能性があるように, 劣勾配を用いて次のように v を更新する.

$$v_n^k := v_n^k + \theta_l s_n^k \quad \forall k \in K, n \in N \quad (4.45)$$

ここで, θ_l は l 回目の繰り返しにおけるステップサイズである. 適当な UND の最良の上界値を UB , l 回目の繰り返しにおける LGB の最適値である UND の下界値を LB_l , パラメータを $\rho (0 < \rho < 2)$ としたときに, θ_l は次式で与えられる.

$$\theta_l = \frac{\rho(UB - LB_l)}{\sum_{k \in K} \sum_{n \in N} (v_n^k)^2} \quad (4.46)$$

上界値 UB が Lagrange 緩和問題の最適値に一致し, $\lim_{l \rightarrow \infty} \theta_l \rightarrow 0$ かつ $\sum_{l=1}^{\infty} \theta_l \rightarrow \infty$ であれば, Lagrange 乗数は最適値に収束する [54]. 一般的に, UB には最適値の代わりに Lagrange ヒューリスティックなどの近似解法による上界値を用いる. $UB \geq$ 最適値であるため, 適当な繰り返し毎に (4.46) 式における ρ を減少させる必要がある.

4.3.3 Lagrange 緩和法

$UAFB$ に対する Lagrange 緩和法の流れをまとめておく.

Lagrange 緩和法

[ステップ 1] 適当な v の初期値を与える. 繰り返し回数の上限を l_{max} , 収束判定基準を ϵ とする. 劣勾配法のパラメータを ρ , ρ の変更率を λ , 変更周期を l_ρ とする. 最良の上界値を UB , 最良の下界値を LB とする. $UB := \infty$, $LB := 0$, $\rho := 1$, $l := 1$ とする.

[ステップ 2] LGB を解き, l 回目の下界値 LB_l , および最適解 \hat{y}, \hat{x} を求める. $LB_l > LB$ であれば $LB := LB_l$ とする.

[ステップ 3] \hat{y} を用いて Lagrange ヒューリスティックを行い, l 回目の上界値 UB_l を求める. $UB_l < UB$ であれば $UB := UB_l$ とする.

[ステップ 4] $l = l_{max}$ または $(UB - LB)/LB < \epsilon$ であれば終了する.

[ステップ 5] $l \bmod l_\rho = 0$ であれば, $\rho := \rho \times \lambda$ とする.

[ステップ 6] \hat{y} , \hat{x} より劣勾配 s を求め, 劣勾配法により v を更新する. $l := l + 1$ と
して, ステップ 2 へ戻る.

ステップ 3 における Lagrange ヒューリスティックは, 4.6 節に示す方法を用いる.

4.4 強い定式化に対する Lagrange 緩和法

4.4.1 Lagrange 緩和問題とその解法

$UAFS$ に対して, Lagrange 乗数 v を用いて, フロー保存式 (4.7) を Lagrange 緩和した問題 LGS を作成する. LGS は次のように表される.

(LGS)

$$\begin{aligned} \text{最小化} \quad & \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} \\ & + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (4.47) \end{aligned}$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall k \in K_{O^k}^o, k \in K, (i, j) \in A \quad (4.48)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4.49)$$

$$\mathbf{y} \in Y \quad (4.50)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.51)$$

はじめに, 連結条件 (4.50) を取り除いた問題 LGS' を考える. 適当な v を与えると目的関数の第一項 $\sum_{k \in K} (v_{D^k}^k - v_{O^k}^k)$ は定数項として扱えるため, LGS' はアーク (i, j) 毎の独立した次のような問題 LGS_{ij} に分割できる.

(LGS_{ij})

$$\text{最小化} \quad \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} + f_{ij} y_{ij} \quad (4.52)$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^k}^o, k \in K \quad (4.53)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K \quad (4.54)$$

$$y_{ij} \in \{0, 1\} \quad (4.55)$$

始点を n とする品種集合 K_n^o を用いると, $k \in K$ は $n \in N, k \in K_n^o$ と表現でき, $h \in K_{O^k}^o$ は $h \in K_n^o$ と表現できるので, LGS_{ij} は次のような問題 LGS'_{ij} に書き直すことができる.

(LGS'_{ij})

$$\text{最小化 } \sum_{n \in N} \sum_{k \in K_n^o} \{(c_{ij}^k - v_j^k + v_i^k)x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k)x_{ji}^k\} + f_{ij}y_{ij} \quad (4.56)$$

$$\text{条件 } x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_n^o, k \in K_n^o, n \in N \quad (4.57)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K_n^o, n \in N \quad (4.58)$$

$$y_{ij} \in \{0, 1\} \quad (4.59)$$

LGS'_{ij} において, y_{ij} は 0 または 1 のどちらかである. そこで, $y_{ij} = 1$ と $y_{ij} = 0$ の場合に分けて考える.

$y_{ij} = 1$ である場合, $f_{ij}y_{ij}$ は定数項となるので, LGS'_{ij} は次のようなノード n 毎の独立した問題 LGS^{n1}_{ij} に分割できる.

(LGS^{n1}_{ij})

$$\text{最小化 } \sum_{k \in K_n^o} \{(c_{ij}^k - v_j^k + v_i^k)x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k)x_{ji}^k\} \quad (4.60)$$

$$\text{条件 } x_{ij}^k + x_{ji}^h \leq 1 \quad \forall k \in K_n^o, h \in K_n^o \quad (4.61)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K_n^o \quad (4.62)$$

(4.61) 式は, 高々一方向のフローを許す非逆流制約式となる. $i \rightarrow j$ 方向のフローが存在するとき, x_{ij}^k の上限は 1 であり, かつ最小化問題であるため, 目的関数値の最適値は $\sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k)$ となる. 一方, $j \rightarrow i$ 方向のフローが存在するとき, 目的関数値の最適値は $\sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k)$ となる. 最小化問題であるため, これらの小さい方が選ばれる. このため, LGS^{n1}_{ij} の最適値 $\tilde{\phi}_{ij}^{n1}$ は,

$$\tilde{\phi}_{ij}^{n1} = \min \left\{ \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k), \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) \right\} \quad n \in N, (i, j) \in A \quad (4.63)$$

となり, 最適解 \tilde{x} は次のようになる.

$$\tilde{x}_{ij}^k = \begin{cases} 1 & \text{if } \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k) \leq \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) \\ & \text{and } c_{ij}^k - v_j^k + v_i^k < 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.64)$$

$$\tilde{x}_{ji}^k = \begin{cases} 1 & \text{if } \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) < \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k) \\ & \text{and } c_{ji}^k - v_i^k + v_j^k < 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.65)$$

一方, $y_{ij} = 0$ である場合, LGS_{ij} の最適値は 0 である. したがって, LGS_{ij} はフロー変数 x を用いない次のような問題 LGSY_{ij} に帰着できる.

(LGSY_{ij})

$$\text{最小化} \quad \left[\sum_{n \in N} \min \left\{ \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k), \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) \right\} + f_{ij} \right] y_{ij} \quad (4.66)$$

$$\text{条件} \quad y_{ij} \in \{0, 1\} \quad (4.67)$$

以上のことから，連結条件 (4.50) も考慮すると，LGS はフロー変数 x を用いない次のような問題 LGSY に帰着できる．

(LGSY)

$$\begin{aligned} \text{最小化} \quad & \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) \\ & + \sum_{(i,j) \in A} \left[\sum_{n \in N} \min \left\{ \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k), \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) \right\} \right. \\ & \left. + f_{ij} \right] y_{ij} \quad (4.68) \end{aligned}$$

$$\text{条件} \quad y \in Y \quad (4.69)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.70)$$

アークの重み W を (4.68) 式の y の係数とし，次のようにおく．

$$W_{ij} = \sum_{n \in N} \min \left\{ \sum_{k \in K_n^o} \min(0, c_{ij}^k - v_j^k + v_i^k), \sum_{k \in K_n^o} \min(0, c_{ji}^k - v_i^k + v_j^k) \right\} + f_{ij} \quad \forall (i, j) \in A \quad (4.71)$$

LGSY は，アークの重み W の合計が最小となる連結を求める問題となる．ただし，重みが負となる場合があることに注意する．この問題の解は，最小木問題の解と重みが負であるアークを併せたものとなる．

LGSY の解法

[ステップ 1] $W_{ij} < 0$ であるアーク $(i, j) \in A$ について $\hat{y}_{ij} := 1$ とし，それ以外を $\hat{y}_{ij} := 0$ とする．

[ステップ 2] \hat{y} により連結される成分をノードに縮約する．連結成分に含まれるノード集合 M の縮約ノード m と G の縮約ノード g 間のアークの重みを $W'_{mg} := \min_{i \in M, j \in G} W_{ij}$ とする．

[ステップ 3] アークの重みを W' とした縮約したネットワーク上で，最小木を求める．

[ステップ 4] 最小木に含まれるアーク (i, j) について， $\hat{y}_{ij} := 1$ とする．

ステップ 3 において，最小木は Kruskal 法 [53] などを用いて求めることができる．

LGSY の最適解 \hat{y} を用いると，LGS の最適解 \hat{x} は次のようになる．

$$\hat{x}_{ij}^k = \begin{cases} \tilde{x}_{ij}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \quad \hat{x}_{ji}^k = \begin{cases} \tilde{x}_{ji}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.72)$$

\hat{y} を $LGSY$ の目的関数に代入すれば, LGS の最適値, すなわち UND の下界値を求めることができる.

4.4.2 劣勾配法

\hat{y}, \hat{x} が求められたときに, LGS において Lagrange 乗数 v を変数と見なした問題は LDB に一致する. そのため, LGS に対する劣勾配法は LGB に対する劣勾配法と同様の手順となる.

v に対して劣勾配法を適用する. v に関する劣勾配 s を v の係数, すなわち,

$$s_n^k = \begin{cases} -1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = O^k \\ 1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = D^k \\ - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{otherwise} \end{cases} \quad \forall k \in K, n \in N \quad (4.73)$$

とする.

この劣勾配を用いて次のように乗数を更新する.

$$v_n^k := v_n^k + \theta_l s_n^k \quad \forall k \in K, n \in N \quad (4.74)$$

ここで, θ_l は l 回目の繰り返しにおけるステップサイズである. 適当な UND の最良の上界値を UB , l 回目の繰り返しにおける LGB の最適値である UND の下界値を LB_l , パラメータを $\rho (0 < \rho < 2)$ としたときに, θ_l は次式で与えられる.

$$\theta_l = \frac{\rho(UB - LB_l)}{\sum_{k \in K} \sum_{n \in N} (v_n^k)^2} \quad (4.75)$$

4.4.3 Lagrange 緩和法

$UAFS$ に対する Lagrange 緩和法の流れをまとめておく.

Lagrange 緩和法

- [ステップ 1] 適当な v の初期値を与える. 繰り返し回数の上限を l_{max} , 収束判定基準を ϵ とする. 劣勾配法のパラメータを ρ , ρ の変更率を λ , 変更周期を l_ρ とする. 最良の上界値を UB , 最良の下界値を LB とする. $UB := \infty$, $LB := 0$, $\rho := 1$, $l := 1$ とする.
- [ステップ 2] LGS を解き, l 回目の下界値 LB_l , および最適解 \hat{y}, \hat{x} を求める. $LB_l > LB$ であれば $LB := LB_l$ とする.
- [ステップ 3] \hat{y} を用いて適当な Lagrange ヒューリスティックを行い, l 回目の上界値 UB_l を求める. $UB_l < UB$ であれば $UB := UB_l$ とする.
- [ステップ 4] $l = l_{max}$ または $(UB - LB)/LB < \epsilon$ であれば終了する.
- [ステップ 5] $l \bmod l_\rho = 0$ であれば, $\rho := \rho \times \lambda$ とする.
- [ステップ 6] \hat{y}, \hat{x} より劣勾配 s を求め, 劣勾配法により v を更新する. $l := l + 1$ として, ステップ 2 へ戻る.

ステップ3における Lagrange ヒューリスティックは、4.6 節に示す方法を用いる。

4.5 拡張した定式化に対する Lagrange 緩和法

4.5.1 Lagrange 緩和問題とその解法

$UAFE$ は膨大な数の変数と制約式を含む組合せ最適化問題となる。そこで、強制制約式としては (4.14) 式を残し、フロー保存式 (4.13) および強制制約式 (4.15) ~ (4.17) を Lagrange 緩和した問題を考える。

(4.13) 式に対する Lagrange 乗数を v とし、(4.15)、(4.16)、(4.17) 式に対する非負の Lagrange 乗数を t, u, r とし、Lagrange 緩和問題 LGE を作成する。はじめに、 LGE の目的関数値 ϕ を整理する。

$$\begin{aligned}
\phi &= \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} y_{ij} \\
&+ \sum_{k \in K} v_{O^k}^k \left(-1 - \sum_{i \in N_{O^k}} x_{iD^k}^k + \sum_{j \in N_{O^k}} x_{O^k j}^k \right) + \sum_{k \in K} v_{D^k}^k \left(1 - \sum_{i \in N_{D^k}} x_{iD^k}^k + \sum_{j \in N_{D^k}} x_{D^k j}^k \right) \\
&+ \sum_{k \in K} \sum_{n \in N \setminus \{O^k, D^k\}} v_n^k \left(- \sum_{i \in N_n} x_{in}^k + \sum_{j \in N_n} x_{nj}^k \right) + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{D^k}^d} t_{ij}^{kh} (x_{ij}^k + x_{ji}^h - y_{ij}) \\
&+ \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{O^k}^d} u_{ij}^{kh} (x_{ij}^k + x_{ji}^h - y_{ij}) + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{O^k}^d} r_{ij}^{kh} (x_{ji}^k + x_{ji}^h - y_{ij}) \\
&= \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} \\
&\quad + \sum_{(i,j) \in A} \left(\sum_{k \in K} \sum_{h \in K_{D^k}^d} t_{ij}^{kh} x_{ij}^k + \sum_{k \in K} \sum_{h \in K_{D^k}^d} t_{ij}^{kh} x_{ji}^h + \sum_{k \in K} \sum_{h \in K_{O^k}^d} u_{ij}^{kh} x_{ij}^k \right. \\
&\quad \left. + \sum_{k \in K} \sum_{h \in K_{O^k}^d} u_{ij}^{kh} x_{ji}^h + \sum_{k \in K} \sum_{h \in K_{O^k}^d} r_{ij}^{kh} x_{ji}^k + \sum_{k \in K} \sum_{h \in K_{O^k}^d} r_{ij}^{kh} x_{ji}^h \right) \\
&\quad + \sum_{(i,j) \in A} \left\{ f_{ij} - \sum_{k \in K} \left(\sum_{h \in K_{D^k}^d} t_{ij}^{kh} + \sum_{h \in K_{O^k}^d} u_{ij}^{kh} + \sum_{h \in K_{O^k}^d} r_{ij}^{kh} \right) \right\} y_{ij} \\
&= \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} \\
&\quad + \sum_{(i,j) \in A} \left(\sum_{k \in K} \sum_{h \in K_{D^k}^d} t_{ij}^{kh} x_{ij}^k + \sum_{h \in K} \sum_{k \in K_{D^h}^d} t_{ij}^{hk} x_{ji}^k + \sum_{k \in K} \sum_{h \in K_{O^k}^d} u_{ij}^{kh} x_{ij}^k \right. \\
&\quad \left. + \sum_{h \in K} \sum_{k \in K_{O^h}^d} u_{ij}^{hk} x_{ij}^k + \sum_{k \in K} \sum_{h \in K_{O^k}^d} r_{ij}^{kh} x_{ji}^k + \sum_{h \in K} \sum_{k \in K_{O^h}^d} r_{ij}^{hk} x_{ji}^k \right) \\
&\quad + \sum_{(i,j) \in A} \left\{ f_{ij} - \sum_{k \in K} \left(\sum_{h \in K_{D^k}^d} t_{ij}^{kh} + \sum_{h \in K_{O^k}^d} u_{ij}^{kh} + \sum_{h \in K_{O^k}^d} r_{ij}^{kh} \right) \right\} y_{ij} \quad (4.76)
\end{aligned}$$

$h \in K, k \in K_{D^d}^k$ を $k \in K, h \in K_{D^d}^k$ に置き換え, $h \in K, k \in K_{O^d}^k$ を $k \in K, h \in K_{D^d}^k$ に置き換えると,

$$\begin{aligned} \phi = & \sum_{k \in K} (v_{D^d}^k - v_{O^d}^k) + \sum_{(i,j) \in A} \sum_{k \in K} \{ (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + (c_{ji}^k - v_i^k + v_j^k) x_{ji}^k \} \\ & + \sum_{(i,j) \in A} \sum_{k \in K} \left\{ \left(\sum_{h \in K_{D^d}^k} t_{ij}^{kh} + \sum_{h \in K_{O^d}^k} u_{ij}^{kh} + \sum_{h \in K_{D^d}^k} u_{ij}^{hk} \right) x_{ij}^k \right. \\ & \quad \left. + \left(\sum_{h \in K_{D^d}^k} t_{ij}^{hk} + \sum_{h \in K_{O^d}^k} r_{ij}^{kh} + \sum_{h \in K_{D^d}^k} r_{ij}^{hk} \right) x_{ji}^k \right\} \\ & + \sum_{(i,j) \in A} \left\{ f_{ij} - \sum_{k \in K} \left(\sum_{h \in K_{D^d}^k} t_{ij}^{kh} + \sum_{h \in K_{O^d}^k} u_{ij}^{kh} + \sum_{h \in K_{D^d}^k} r_{ij}^{kh} \right) \right\} y_{ij} \quad (4.77) \end{aligned}$$

となる. したがって, LGE は次のように表わされる.

(LGE)

$$\text{最小化} \quad \sum_{k \in K} (v_{D^d}^k - v_{O^d}^k) + \sum_{(i,j) \in A} \sum_{k \in K} (\alpha_{ij}^k x_{ij}^k + \beta_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} \gamma_{ij} y_{ij} \quad (4.78)$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_{O^d}^k, k \in K, (i,j) \in A \quad (4.79)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i,j) \in A \quad (4.80)$$

$$\mathbf{y} \in Y \quad (4.81)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \quad (4.82)$$

ここで,

$$\alpha_{ij}^k = c_{ij}^k - v_j^k + v_i^k + \sum_{k \in K} \left(\sum_{h \in K_{D^d}^k} t_{ij}^{kh} + \sum_{h \in K_{O^d}^k} u_{ij}^{kh} + \sum_{h \in K_{D^d}^k} u_{ij}^{hk} \right) \quad (4.83)$$

$$\beta_{ji}^k = c_{ji}^k - v_i^k + v_j^k + \sum_{k \in K} \left(\sum_{h \in K_{D^d}^k} t_{ij}^{hk} + \sum_{h \in K_{O^d}^k} r_{ij}^{kh} + \sum_{h \in K_{D^d}^k} r_{ij}^{hk} \right) \quad (4.84)$$

$$\gamma_{ij} = f_{ij} - \sum_{k \in K} \left\{ \sum_{h \in K_{D^d}^k} t_{ij}^{kh} + \sum_{h \in K_{O^d}^k} (u_{ij}^{kh} + r_{ij}^{kh}) \right\} \quad (4.85)$$

である.

はじめに, 連結条件 (4.81) を取り除いた問題 LGE' を考える. Lagrange 乗数が与えられたとき, 目的関数の第一項は定数と見なすことができる. そのため, LGE' は次のようなアーク (i,j) 毎の問題 LGE_{ij} に分割することができる.

(LGE_{ij})

$$\text{最小化} \quad \sum_{n \in N} \sum_{k \in K_n^o} (\alpha_{ij}^k x_{ij}^k + \beta_{ji}^k x_{ji}^k) + \gamma_{ij} y_{ij} \quad (4.86)$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall h \in K_n^o, k \in K_n^o, n \in N \quad (4.87)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K_n^o, n \in N \quad (4.88)$$

$$y_{ij} \in \{0, 1\} \quad (4.89)$$

ここで, $k \in K$ を $n \in N, k \in K_n^o$ に置き換え, $h \in K_{O_k}^o$ を $h \in K_n^o$ に置き換えている.

LGE_{ij} において, y_{ij} は 0 または 1 のどちらかである. そこで, $y_{ij} = 1$ と $y_{ij} = 0$ の場合に分けて考える.

$y_{ij} = 1$ である場合, $\gamma_{ij}y_{ij}$ は定数項となるので, LGE_{ij} は次のような品種の始点であるノード n 毎の独立した問題 LGE_{ij}^{n1} に分割できる.

(LGE_{ij}^{n1})

$$\text{最小化} \quad \sum_{k \in K_n^o} (\alpha_{ij}^k x_{ij}^k + \beta_{ji}^k x_{ji}^k) \quad (4.90)$$

$$\text{条件} \quad x_{ij}^k + x_{ji}^h \leq 1 \quad \forall h \in K_n^o, k \in K_n^o \quad (4.91)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K_n^o \quad (4.92)$$

LGE_{ij}^{n1} は, 上限が 1 である制約をもつ問題である. そのため, $x_{ij}^k > 0$ であるためには, その係数が負であることが必要である. (4.91) 式は非逆流不等式であり, $i \rightarrow j$ と $j \rightarrow i$ 方向のどちらか高々一方のフローの存在のみを許す制約式となる. $i \rightarrow j$ 方向のフローが存在するためには $\sum_{m \in K_n^o} \min(0, \alpha_{ij}^m) \leq \sum_{m \in K_n^o} \min(0, \beta_{ji}^m)$ であることが必要であり, $j \rightarrow i$ 方向のフローが存在するためには $\sum_{m \in K_n^o} \min(0, \beta_{ji}^m) < \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m)$ であることが必要である. 最小化問題であるため, これらの小さい方が選ばれる.

したがって, LGE_{ij}^{n1} の最適値 $\tilde{\phi}_{ij}^{n1}$ は,

$$\tilde{\phi}_{ij}^{n1} = \min \left\{ \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m), \sum_{m \in K_n^o} \min(0, \beta_{ji}^m) \right\} \quad n \in N, (i, j) \in A \quad (4.93)$$

となり, LGE_{ij}^{n1} の最適解 \tilde{x} は, 次のようになる.

$$\tilde{x}_{ij}^k = \begin{cases} 1 & \text{if } \alpha_{ij}^k < 0 \text{ and } \sum_{m \in K_{O_k}^o} \min(0, \alpha_{ij}^m) \leq \sum_{m \in K_{O_k}^o} \min(0, \beta_{ji}^m) \\ 0 & \text{otherwise} \end{cases} \quad k \in K, (i, j) \in A \quad (4.94)$$

$$\tilde{x}_{ji}^k = \begin{cases} 1 & \text{if } \beta_{ji}^k < 0 \text{ and } \sum_{m \in K_{O_k}^o} \min(0, \beta_{ji}^m) < \sum_{m \in K_{O_k}^o} \min(0, \alpha_{ij}^m) \\ 0 & \text{otherwise} \end{cases} \quad k \in K, (i, j) \in A \quad (4.95)$$

一方, $y_{ij} = 0$ である場合, LGE_{ij} の最適値は 0 である. したがって, LGE_{ij} はアークフロー変数を用いない次のような問題 $LGEY_{ij}$ に置き換えることができる.

($LGEY_{ij}$)

$$\text{最小化} \quad \left[\sum_{n \in N} \min \left\{ \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m), \sum_{m \in K_n^o} \min(0, \beta_{ji}^m) \right\} + \gamma_{ij} \right] y_{ij} \quad (4.96)$$

$$\text{条件} \quad y_{ij} \in \{0, 1\} \quad (4.97)$$

以上のことから，連結条件 (4.81) も考慮すると， LGE は次のような問題 $LGEY$ に帰着される．

($LGEY$)

$$\begin{aligned} \text{最小化} \quad & \sum_{k \in K} (v_{D^k}^k - v_{O^k}^k) \\ & + \sum_{(i,j) \in A} \left[\sum_{n \in N} \min \left\{ \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m), \sum_{m \in K_n^o} \min(0, \beta_{ji}^m) \right\} + \gamma_{ij} \right] y_{ij} \quad (4.98) \end{aligned}$$

$$\text{条件 } \mathbf{y} \in Y \quad (4.99)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.100)$$

ここで，アークの重み W を (4.98) 式の y_{ij} の係数とし，次のようにおく．

$$W_{ij} = \sum_{n \in N} \min \left\{ \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m), \sum_{m \in K_n^o} \min(0, \beta_{ji}^m) \right\} + \gamma_{ij} \quad \forall (i, j) \in A \quad (4.101)$$

$LGEY$ は，アークの重み W の合計が最小となる連結を求める問題となる．この問題の解は，最小木問題の解と重みが負であるアークを併せたものとなる． $LGEY$ は $LGSY$ と同様に最小木問題に帰着され，Kruskal 法などで容易に解くことができる．

$LGEY$ の最適解 \hat{y} は LGE の最適解となる． \hat{y} を (4.98) 式に代入すれば， LGE の最適値，すなわち UND の下界値を求めることができる．また， LGE の最適解 \hat{x} は次のようになる．

$$\hat{x}_{ij}^k = \begin{cases} \hat{x}_{ij}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \quad \hat{x}_{ji}^k = \begin{cases} \hat{x}_{ji}^k & \text{if } \hat{y}_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.102)$$

4.5.2 Lagrange 乗数の設定と劣勾配法

LGE の最適値は UND の下界値であり，この下界値を最大にするように Lagrange 乗数を設定することが必要である． LGE の最適解 \hat{y} ， \hat{x} が求められているとき，Lagrange 乗数 v, t, u, w に関する最適化問題 LDE は次のようになる．

(LDE)

$$\begin{aligned} \text{最大化} \quad & \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k \hat{x}_{ij}^k + c_{ji}^k \hat{x}_{ji}^k) + \sum_{(i,j) \in A} f_{ij} \hat{y}_{ij} \\ & \sum_{k \in K} v_{O^k}^k \left(-1 - \sum_{i \in N_{O^k}} \hat{x}_{iO^k}^k + \sum_{j \in N_{O^k}} \hat{x}_{O^k j}^k \right) + \sum_{k \in K} v_{D^k}^k \left(1 - \sum_{i \in N_{D^k}} \hat{x}_{iD^k}^k + \sum_{j \in N_{D^k}} \hat{x}_{D^k j}^k \right) \\ & \sum_{k \in K} \sum_{n \in N \setminus \{D^k, O^k\}} v_n^k \left(- \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k \right) + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{D^k}^d} t_{ij}^{kh} \left(\hat{x}_{ij}^k + \hat{x}_{ji}^h - \hat{y}_{ij} \right) \\ & + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{O^k}^d} u_{ij}^{kh} \left(\hat{x}_{ij}^k + \hat{x}_{ji}^h - \hat{y}_{ij} \right) + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{h \in K_{O^k}^d} r_{ij}^{kh} \left(\hat{x}_{ji}^k + \hat{x}_{ji}^h - \hat{y}_{ij} \right) \quad (4.103) \end{aligned}$$

$$\text{条件 } t_{ij}^{kh} \geq 0 \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (4.104)$$

$$u_{ij}^{kh} \geq 0 \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.105)$$

$$r_{ij}^{kh} \geq 0 \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.106)$$

\hat{y}, \hat{x} は組合せ最適化問題 *LGE* の解であり, Lagrange 乗数が変化したときに離散的に変化するため, *LDE* の目的関数値も離散的に変化する. このため *LDE* は微分不可能な目的関数をもつ最適化問題となり, 微分不可能な問題に対する解法が必要となる.

v に対しては, 劣勾配法を適用する. *LDE* の目的関数は v の 1 次関数であることから, v に関する劣勾配 s を v の係数, すなわち,

$$s_n^k = \begin{cases} -1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = O^k \\ 1 - \sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{if } n = D^k \\ -\sum_{i \in N_n} \hat{x}_{in}^k + \sum_{j \in N_n} \hat{x}_{nj}^k & \text{otherwise} \end{cases} \quad \forall k \in K, (i, j) \in A \quad (4.107)$$

とする.

一方, t, u, r は多数存在するため, すべてを陽的に考慮することは困難である. そのため, 逐次, \hat{y}, \hat{x} が満足しない (4.15) ~ (4.17) 式, すなわち,

$$\hat{x}_{ij}^k + \hat{x}_{ji}^h > \hat{y}_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.108)$$

$$\hat{x}_{ij}^k + \hat{x}_{ij}^h > \hat{y}_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.109)$$

$$\hat{x}_{ji}^k + \hat{x}_{ji}^h > \hat{y}_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.110)$$

である制約式に対してのみ, Lagrange 乗数に非負の値を設定し, それ以外は 0 とする.

$\hat{y}_{ij} = 0$ である場合, $\hat{x}_{ij}^k = \hat{x}_{ji}^h = 0$ となる. このため, (4.108) 式が成り立つのは, $\hat{y}_{ij} = 1$ かつ $\hat{x}_{ij}^k = \hat{x}_{ji}^h = 1$ の場合に限られる.

$$\zeta_{ij}^n = \sum_{m \in K_n^o} \min(0, \alpha_{ij}^m), \quad \eta_{ji}^n = \sum_{m \in K_n^o} \min(0, \beta_{ji}^m) \quad \forall n \in N, (i, j) \in A \quad (4.111)$$

とおくと, $\hat{x}_{ij}^k = 1$ であるためには $\alpha_{ij}^k < 0$ かつ $\zeta_{ij}^{O^k} \leq \eta_{ji}^{O^k}$ であることが必要となる. また, $\hat{x}_{ji}^h = 1$ であるためには $\beta_{ji}^h < 0$ かつ $\eta_{ji}^{O^h} < \zeta_{ij}^{O^h}$ であることが必要となる. したがって, ある k, h に対して (4.108) 式が成り立つためには,

$$\hat{y}_{ij} = 1 \text{ and } \max(\alpha_{ij}^k, \beta_{ji}^h, \zeta_{ij}^{O^k} - \eta_{ji}^{O^k}, \eta_{ji}^{O^h} - \zeta_{ij}^{O^h}) < 0 \quad (4.112)$$

であることが必要となる. そこで, *LGE* の最適解 \hat{y}, \hat{x} が求められたとき, はじめて (4.112) 式を満足する $((i, j), k, h)$ に対応する強制制約式 (4.15) の Lagrange 乗数 t_{ij}^{kh} の値を次のように設定する.

$$t_{ij}^{kh} = -\max(\alpha_{ij}^k, \beta_{ji}^h, \zeta_{ij}^{O^k} - \eta_{ji}^{O^k}, \eta_{ji}^{O^h} - \zeta_{ij}^{O^h}) \quad (4.113)$$

このとき, t_{ij}^{kh} が関係する $\alpha_{ij}^k, \beta_{ji}^h, \zeta_{ij}^{O^k}, \eta_{ji}^{O^h}$ が変化するため,

$$\alpha_{ij}^k := \alpha_{ij}^k + t_{ij}^{kh} \quad (4.114)$$

$$\beta_{ji}^h := \beta_{ji}^h + t_{ij}^{kh} \quad (4.115)$$

$$\zeta_{ij}^{O^k} := \zeta_{ij}^{O^k} + t_{ij}^{kh} \quad (4.116)$$

$$\eta_{ji}^{O^h} := \eta_{ji}^{O^h} + t_{ij}^{kh} \quad (4.117)$$

と更新する.

同様に, (4.109) 式, (4.110) 式が成り立つためには, 次式が成り立つことが必要となる.

$$\hat{y}_{ij} = 1 \text{ and } \max(\alpha_{ij}^k, \alpha_{ij}^h, \zeta_{ij}^{O^k} - \eta_{ji}^{O^k}, \zeta_{ij}^{O^h} - \eta_{ji}^{O^h}) < 0 \quad (4.118)$$

$$\hat{y}_{ij} = 1 \text{ and } \max(\beta_{ji}^k, \beta_{ji}^h, \eta_{ji}^{O^k} - \zeta_{ij}^{O^k}, \eta_{ji}^{O^h} - \zeta_{ij}^{O^h}) < 0 \quad (4.119)$$

はじめて (4.118) 式を満足する $((i, j), k, h)$ に対応する強制制約式 (4.16) のLagrange乗数 u_{ij}^{kh} の値を

$$u_{ij}^{kh} = -\max(\alpha_{ij}^k, \alpha_{ij}^h, \zeta_{ij}^{O^k} - \eta_{ji}^{O^k}, \zeta_{ij}^{O^h} - \eta_{ji}^{O^h}) \quad (4.120)$$

とし, 次のように更新する.

$$\alpha_{ij}^k := \alpha_{ij}^k + u_{ij}^{kh} \quad (4.121)$$

$$\alpha_{ij}^h := \alpha_{ij}^h + u_{ij}^{kh} \quad (4.122)$$

$$\zeta_{ij}^{O^k} := \zeta_{ij}^{O^k} + u_{ij}^{kh} \quad (4.123)$$

$$\zeta_{ij}^{O^h} := \zeta_{ij}^{O^h} + u_{ij}^{kh} \quad (4.124)$$

さらに, はじめて (4.119) 式を満足する $((i, j), k, h)$ に対応する強制制約式 (4.17) のLagrange乗数 r_{ij}^{kh} の値を

$$r_{ij}^{kh} = -\max(\beta_{ji}^k, \beta_{ji}^h, \eta_{ji}^{O^k} - \zeta_{ij}^{O^k}, \eta_{ji}^{O^h} - \zeta_{ij}^{O^h}) \quad (4.125)$$

とし, 次のように更新する.

$$\beta_{ji}^k := \beta_{ji}^k + r_{ij}^{kh} \quad (4.126)$$

$$\beta_{ji}^h := \beta_{ji}^h + r_{ij}^{kh} \quad (4.127)$$

$$\eta_{ji}^{O^k} := \eta_{ji}^{O^k} + r_{ij}^{kh} \quad (4.128)$$

$$\eta_{ji}^{O^h} := \eta_{ji}^{O^h} + r_{ij}^{kh} \quad (4.129)$$

非負である乗数に対してのみ, 劣勾配法を適用する. Lagrange乗数 t, u, r に対する劣勾配 p, q, g には (4.103) における係数を用いる.

$$p_{ij}^{kh} = x_{ij}^k + x_{ji}^h - y_{ij} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (4.130)$$

$$q_{ij}^{kh} = x_{ij}^k + x_{ij}^h - y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.131)$$

$$g_{ij}^{kh} = x_{ji}^k + x_{ji}^h - y_{ij} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.132)$$

LDE の目的関数値を改善する可能性があるように，劣勾配を用いて次のように乗数を更新する．

$$v_n^k := v_n^k + \theta_l s_n^k \quad \forall k \in K, n \in N \quad (4.133)$$

$$t_{ij}^{kh} := \begin{cases} \max(0, t_{ij}^{kh} + \theta_l p_{ij}^{kh}) & \text{if } t_{ij}^{kh} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in K_{D^k}^d, k \in K, (i, j) \in A \quad (4.134)$$

$$u_{ij}^{kh} := \begin{cases} \max(0, u_{ij}^{kh} + \theta_l q_{ij}^{kh}) & \text{if } u_{ij}^{kh} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.135)$$

$$r_{ij}^{kh} := \begin{cases} \max(0, r_{ij}^{kh} + \theta_l g_{ij}^{kh}) & \text{if } r_{ij}^{kh} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in K_{O^k}^d, k \in K, (i, j) \in A \quad (4.136)$$

ここで， θ_l は l 回目の繰り返しにおけるステップサイズである．適当な UND の最良の上界値を UB ， l 回目の繰り返しにおける LGE の最適値である UND の下界値を LB_l ，パラメータを $\rho (0 < \rho < 2)$ としたときに， θ_l は次式で与えられる．

$$\theta_l = \frac{\rho(UB - LB_l)}{\sum_{k \in K} \left[\sum_{n \in N} (v_n^k)^2 + \sum_{(i,j) \in A} \left\{ \sum_{h \in K_{D^k}^d} (t_{ij}^{kh})^2 + \sum_{h \in K_{O^k}^d} ((u_{ij}^{kh})^2 + (r_{ij}^{kh})^2) \right\} \right]} \quad (4.137)$$

上界値 UB が緩和問題の最適値に一致し， $\lim_{k \rightarrow \infty} \theta_l \rightarrow 0$ かつ $\sum_{l=1}^{\infty} \theta_l \rightarrow \infty$ であれば，Lagrange 乗数は最適値に収束する．

4.5.3 Lagrange 緩和法

繰り返し毎に，(4.108)~(4.110) 式を満足する $((i, j), k, h)$ に対する強制制約式の Lagrange 乗数値をすべて設定してしまうと，膨大な数の乗数が非負となる．初期の繰り返しでは， v の値がその最適値から大きく離れており，また変化量も大きいため，最適解において不要であろう強制制約式に対する Lagrange 乗数値を設定してしまう可能性がある．このため，繰り返しが一定回数を超えて， v の変化量が小さくなってから，強制制約式に対する Lagrange 乗数値を設定する．その後，適当な繰り返しの周期毎に新たに Lagrange 乗数値を設定する．これにより，新たに正の値を設定する Lagrange 乗数の数を制御し，時間計算量と空間計算量を抑える．

$UAFE$ に対する Lagrange 緩和法の流れをまとめておく．

Lagrange 緩和法

[ステップ 1] 適当な v の初期値を与える．繰返し回数の上限を l_{max} ，収束判定基準を ϵ とする．劣勾配法のパラメータを ρ ， ρ の変更率を λ ，変更周期を l_ρ とする． t ， u ， r の値の設定を開始できる繰返し回数を l_{str} ，生成する周期を l_{int} とする．最良の上界値を UB ，最良の下界値を LB とする． $UB := \infty$ ， $LB := 0$ ， $l := 1$ ， $\rho := 1$ ， $t = 0$ ， $u = 0$ ， $r = 0$ とする．

[ステップ 2] LGE を解き， l 回目の下界値 LB_l ，および最適解 \hat{y} ， \hat{x} を求める． $LB_l > LB$ であれば $LB := LB_l$ とする．

[ステップ 3] $l \geq l_{str}$ かつ $l \bmod l_{int} = 0$ であれば，(4.108)~(4.110) 式を初めて満足

する $((i, j), k, h)$ に対する強制制約式の Lagrange 乗数 t, u, r の値を設定する .

[ステップ 4] \hat{y} を用いて適当な Lagrange ヒューリスティックを行い, l 回目の上界値 UB_l を求める . $UB_l < UB$ であれば $UB := UB_l$ とする .

[ステップ 5] $l = l_{max}$ または $(UB - LB)/LB < \epsilon$ であれば終了する .

[ステップ 6] $l \bmod l_\rho = 0$ であれば, $\rho := \rho \times \lambda$ とする .

[ステップ 7] \hat{y} および \hat{x} より劣勾配 s, p, q, g を求め, 劣勾配法により v, t, u, r を更新する . $l := l + 1$ として, ステップ 2 へ戻る .

ステップ 4 における Lagrange ヒューリスティックには, 4.6 節に示す方法を用いる .

4.6 Lagrange ヒューリスティック

Lagrange 緩和解を用いた三種類の Lagrange ヒューリスティックを示す . Lagrange 緩和法では, 劣勾配法の繰返し毎に緩和解が得られる . Lagrange ヒューリスティックは, Lagrange 緩和解を改良して近似解および上界値を求める解法である . 劣勾配法の繰返し毎に近似解を求めることができ, これらの内の最良解を最終的な近似解として採用する .

初期解

前節までに示した Lagrange 緩和問題ではフロー保存式を緩和しているため, フロー変数がフロー保存式を満たしているとは限らない . しかし, デザイン変数解は, $UAFB$ では連結条件を満たせば実行可能解となり, $UAFS$ および $UAFE$ では連結条件を満たしているため実行可能解となる .

Lagrange 緩和問題のデザイン変数解を \hat{y} とする . $UAFB$ において y を \hat{y} に固定した問題 $AFY(\hat{y})$ は次のようになる .

($AFY(\hat{y})$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k) + \sum_{(i,j) \in A} f_{ij} \hat{y}_{ij} \quad (4.138)$$

$$\text{条件} \quad \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (4.139)$$

$$x_{ij}^k + x_{ji}^k \leq \hat{y}_{ij} \quad \forall k \in K, (i, j) \in A \quad (4.140)$$

$$x_{ji}^k \leq \hat{y}_{ij} \quad \forall k \in K, (i, j) \in A \quad (4.141)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4.142)$$

目的関数の末項は定数項であるので, $AFY(\hat{y})$ は品種 k 毎の独立した問題 $AFY^k(\hat{y})$ に分解することができる . $AFY^k(\hat{y})$ は, $\hat{y}_{ij} = 1$ であるアークにより構成されたネットワーク上において, アーク (i, j) の長さを c_{ij}^k , アーク (j, i) の長さを c_{ji}^k とした品種 k の始点・終点間の最短経路問題となるため, これらの問題を解けば $UAFB$ のフロー変数の実行可能解 \hat{x} を求めることができる .

UAFS および *UAFE* において, \hat{y} は実行可能解ではあるが, 最適解ではない. *LGBY* では \hat{y} が実行可能解ではない可能性がある. 実行可能なデザイン変数解を用いて, 実行可能なフロー変数が求まれば, これらは *UND* の実行可能解となる. さらに, デザイン変数解を初期解とした貪欲法などを適用すれば, より良い解を求めることができる.

限定した Minoux タイプのフォワード法

フォワード法は, 初期解より, 目的関数値が減少するようなアークを逐次加えていく貪欲法 [48] である. ここでは, Lagrange 緩和解におけるデザイン変数解 \hat{y} を初期解として, 限定した Minoux タイプのリストを用いたアークを付加していくフォワード法を適用する.

LGBY(または *LGSY*, *LGEY*) における y_{ij} の係数が負であれば, $y_{ij} = 1$ が解となる. そのため, y_{ij} の係数が小さい方が最適解に含まれる可能性が高くなる. そこで, *LGBY*(または *LGSY*, *LGEY*) における y_{ij} の係数を昇順に並べ換え, 係数の小さい順に κ 本を選択し, これらのアークのデザイン変数を付加するアークの候補とする. κ を大きくすれば, 最適解において 1 となるアークを近似解に含めることができる可能性が高くなる. このように, 付加するアークを限定することにより, 計算量を抑える.

各アークに対して, Lagrange 緩和解による初期解においてアークを付加したときの目的関数値の減少量を計算し, 目的関数値の減少量とアークを目的関数値の減少量の降順にリストに格納しておく. なお, 目的関数値の減少量は Murchland 法によって求める. 目的関数値の減少量の大きい順にアークの付加を検討する. アークを付加する対象となったときに, 現在のネットワークにおいて, 目的関数値の減少量を再計算する. この減少量が正でリスト内で最大であるときのみアークを付加する. この減少量が正でリスト内で最大でないときには, リストの適切な位置に目的関数値の減少量とアークを格納する.

Minoux タイプのフォワード法

[ステップ 1] Lagrange 緩和解より, 初期解を求める. 付加対象基準 κ により, 付加対象のアークの候補を選定する.

[ステップ 2] すべての付加対象アーク (i, j) について, このアークを付加した場合の目的関数値の減少量 $\hat{\Delta}_{ij}$ を計算する.

[ステップ 3] $\hat{\Delta}_{ij} > 0$ であるすべてのアーク (i, j) について, $\hat{\Delta}_{ij}$ の降順に該当するアークと $\hat{\Delta}_{ij}$ をリストに格納する.

[ステップ 4] リスト内に $\hat{\Delta}_{ij} > 0$ であるアーク (i, j) がなければ, 終了する. そうでなければ, リストから $\hat{\Delta}_{ij}$ が最大であるアークを取り出す. このアークを (p, q) とする.

[ステップ 5] アーク (p, q) に対して, 現在のネットワークにおける $\hat{\Delta}_{pq}$ を再計算する.

- $\hat{\Delta}_{pq} > 0$ で, リスト内で最大であれば, ステップ 6 へ.
- $\hat{\Delta}_{pq} > 0$ であれば, $\hat{\Delta}_{pq}$ の値にしたがって, アーク (p, q) と $\hat{\Delta}_{pq}$ をリスト内の適切な位置に挿入し, ステップ 4 へ戻る.

- $\hat{\Delta}_{pq} \leq 0$ であれば、ステップ4へ戻る。
- [ステップ6] アーク (p, q) をネットワークに付加し、ステップ4へ戻る。

限定した Minoux 法の改良法2

前章で示した Minoux 法の改良法2は、目的関数値が減少するようなアークを削除していく方法である。ここでは、初期解に含むアークを限定した方法を用いる。

$LGBY$ (または $LGSY$, $LGEY$) における y_{ij} の係数を昇順に並べ換え、係数の小さい順に κ 本を選択し、これらのアークのデザイン変数を1とする。加えて、 $\hat{y}_{ij} = 1$ であるアークのデザイン変数を1としたものを初期解とする。 κ を大きくすれば、最適解において1となるアークを初期解に含めることができる可能性が高くなる。

続いて、Minoux 法の改良法2を用いて、初期解を改善する。Minoux 法の改良法は高速な解法であり、初期解に含まれるアーク数は高々 $\kappa + |N - 1|$ 本と限定されるため、短時間で改善した解を求めることができる。

ω 近傍局所探索法

前述の二つの近似解法によって、現在までの最良の上界値と $\psi\%$ 以内の差の上界値が求まった場合、局所探索法により、これらを初期解として解を改善する。局所探索法は、現在の解に含まれていないアークを付加し、現在の解に含まれているアークを削除したときの目的関数値の変化量を計算し、目的関数値が最も減少するようなアークの付加と削除を同時に行うアーク交換を行い、この操作を目的関数値が減少する間、繰り返す方法である。アーク交換の組合せ数は非常に多いため、付加を検討するアークは、削除するアークの端点の近傍にある ω 個のノードを端点とするアークに限定する。これにより、計算量を抑える。このような方法を ω 近傍局所探索法とよぶ。

Lagrange ヒューリスティック

Lagrange ヒューリスティックの手順は、次のようになる。

- Lagrange ヒューリスティック**
- [ステップ1] アークの選択基準を κ, ω とし、局所探索法実施基準を ψ とする。 $\bar{y} := 0$ とする。
- [ステップ2] Lagrange 緩和問題 LGB (または LGS , LGE) を解き、最適解 \hat{y} を求める。アーク (i, j) において、 $\hat{y}_{ij} = 1$ であれば $\bar{y}_{ij} = 1$ とする。
- [ステップ3] $LGBY$ (または $LGSY$, $LGEY$) における y_{ij} の係数を昇順に並べ換える。係数の小さい順に κ 本のアーク (i, j) を選択し、 $\bar{y}_{ij} = 1$ とする。 \bar{y} で構成されるネットワークが連結しなければ終了する。
- [ステップ4] $\hat{y}_{ij} = 1$ であるアーク (i, j) で構成されるネットワークを初期解とする。 $\hat{y}_{ij} = 0$ かつ $\bar{y}_{ij} = 1$ であるアークに対して限定した Minoux タイプのフォワード法を適用し、近似解を求める。

- [ステップ 5] \bar{y} を初期解とした限定した Minoux 法の改良法 2 を適用する .
- [ステップ 6] ステップ 4 またはステップ 5 で求められた上界値が , 現在までの最良の上界値と $\psi\%$ 以内であれば , これらの近似解を初期解とした ω 近傍局所探索法を行う .
- [ステップ 7] ステップ 4~6 で求められた内の最良解を近似解とする .

ステップ 4 と 5 では , ネットワーク上で連結したときのみ近似解を求めている . $UAFS$ および $UAFE$ では必ず初期解が求まるが , $UAFB$ では初期解が求まらない場合がある .

4.7 数値実験

数値実験によって , 従来の解法と本章で提案した解法の比較を行う . 数値実験には , アーク上のフロー費用とデザイン費用が比例するようなユークリッド平面上の問題例を用いる . なお , 使用したデータは www.rku.ac.jp/~katayama/networkdata に掲載している .

主な条件およびパラメータは次の通りである .

- 1) ノード : 1000 × 1000 のユークリッド平面上にランダムに発生させる .
- 2) アーク : 2 ノード間に設定する . ノード間のユークリッド距離の短いものをアーク候補とする . ただし , 2 ノード間の距離を重みとした最小木に含まれるアークを含む .
- 3) 品種 : 品種に対応する始点・終点の対をランダムに選定する . ただし , 各ノードを始点または終点とする品種が少なくとも 1 品種は存在するものとする .
- 4) 問題数 : 同一ノード数について , 各 5 問とする .
- 5) ノード数 : 10 ~ 100 とする . これらの問題を N10 問題 , \dots , N100 問題とする .
- 6) アーク数 : ノード数の 5 倍とする .
- 7) 品種数 : ノード数の 5 倍とする .
- 8) デザイン費用 : ノード間のユークリッド距離に比例する .
- 9) フロー費用 : ノード間のユークリッド距離に比例する .
- 10) デザイン費用/フロー費用の比 : 10 , 20 , 30 とする . これらの問題を C10 問題 , C20 問題 , C30 問題とする .
- 11) パラメータ
 - 収束判定基準 ϵ : 0.01% とする .
 - ρ の減少間隔 l_ρ : 20 回とする .
 - ρ の減少率 λ : $UAFB$ と $UAFS$ は 0.98 , $UAFE$ は 0.99 とする .
 - 劣勾配法の繰返し回数 l_{max} : $UAFB$ と $UAFS$ は 10000 回 , $UAFE$ は 15000 回とする .
 - $UAFE$ における Lagrange 乗数値の設定開始繰返し回数 l_{str} : 1800 回とする .
 - $UAFE$ における Lagrange 乗数値の設定周期 l_{int} : 100 回とする .

表 4.1 問題の属性 1

問題	デザイン費用・ フロー費用比
C10	10
C20	20
C30	30

表 4.2 問題の属性 2

問題	ノード	アーク	品種
N10	10	45	45
N20	20	100	100
N30	30	150	150
N40	40	200	200
N50	50	250	250
N60	60	300	300
N70	70	350	350
N80	80	400	400
N90	90	450	450
N100	100	500	500

- Lagrange ヒューリスティックにおけるパラメータ κ, ω, ψ : $\kappa =$ ノード数 $\times 10$, $\omega = 10$, $\psi = 0.2\%$ とする.
- 12) 計算時間: Lagrange 緩和法では計算時間の上限を設定しない. CLPEX による計算の上限時間は 10 時間とし, その時点の最良解を採用する.
- 13) 使用計算機: CPU Pentium Core2-Duo 3.3GHz, メモリ 4GB, OS Windows XP
- 14) 使用言語: Intel FORTRAN Ver9.1
- 15) 下界値の誤差: (上界値 - 下界値)/下界値 $\times 100$ の平均値とする.
本章の解法と比較する解法, および第 3 章で比較した解法によって得られた上界値の内の最良値を用いる. なお, C10 問題と C20 問題はノード数 40 まで, C30 問題はノード数 30 までは CPLEX による最適値を用いる.
- 16) 上界値の誤差: (上界値 - 下界値)/下界値 $\times 100$ の平均値とする.
本章の下界値および CPLEX の分枝限定法によって得られた下界値の内の最良値を用いる. なお, C10 問題と C20 問題はノード数 40 まで, C30 問題はノード数 30 までは CPLEX による最適値を用いる.

問題の属性を表 4.1 および表 4.2 に示す. 表 4.1 は, 各問題におけるアーク上のデザイン費用とフロー費用の比であり, 数字が大きいくほどデザイン費用が大きくなる. 表 4.2 は,

表 4.3 下界値の誤差の比較

問題		双対上昇 (%)	<i>UAFB</i> (%)	<i>UAFS</i> (%)	<i>UAFE</i> (%)
C10	N10	2.77	0.50	0.16	0.00
	N20	6.06	0.85	0.48	0.23
	N30	6.99	1.42	0.97	0.69
	N40	7.83	1.66	1.21	0.90
	N50	7.75	1.79	1.35	1.12
	N60	7.84	1.86	1.50	1.27
	N70	8.34	2.10	1.81	1.56
	N80	9.46	2.75	2.44	2.26
	N90	9.00	2.72	2.43	2.24
	N100	8.38	2.36	2.03	1.86
	平均	7.44	1.80	1.44	1.21
C20	N10	8.03	2.10	0.27	0.03
	N20	12.27	2.93	0.34	0.11
	N30	13.36	3.57	0.76	0.48
	N40	14.02	3.89	1.63	1.20
	N50	14.46	4.31	1.85	1.51
	N60	14.95	4.46	2.07	1.75
	N70	14.68	4.36	2.26	1.98
	N80	16.32	5.61	3.23	2.95
	N90	16.38	5.65	3.40	3.18
	N100	15.35	5.07	3.07	2.82
	平均	13.98	4.19	1.89	1.60
C30	N10	13.07	4.26	0.06	0.00
	N20	18.38	5.76	0.20	0.06
	N30	18.35	6.22	0.56	0.32
	N40	20.28	6.46	1.49	1.03
	N50	20.54	7.14	1.74	1.41
	N60	21.40	7.36	2.11	1.75
	N70	20.41	7.04	2.45	2.10
	N80	22.01	8.21	3.08	2.77
	N90	22.72	8.55	3.63	3.38
	N100	21.13	7.67	3.21	2.91
	平均	19.83	6.87	1.85	1.57
平均	13.75	4.29	1.73	1.46	

各問題におけるノード数，アーク数と品種数である．

表 4.3に，従来の解法である双対上昇法 [7] と三種類の Lagrange 緩和法による下界値の上界値との平均誤差を示す．C10 問題と C20 問題はノード数 40 まで，C30 問題はノード数 30 までの上界値は最適値であるため，平均誤差は最適値と下界値との差である．また，それら以外の問題の誤差には，上界値と最適値の差，および最適値と下界値との差の

表 4.4 下界値の計算時間の比較

問題		双対上昇 (秒)	<i>UAFB</i> (秒)	<i>UAFS</i> (秒)	<i>UAFE</i> (秒)
C10	N10	0.0	0.9	1.0	1.9
	N20	0.0	8.3	12.0	30.9
	N30	0.1	20.9	31.1	99.1
	N40	0.3	42.8	63.4	189.4
	N50	0.7	72.3	109.2	295.9
	N60	1.2	113.4	173.0	457.1
	N70	1.7	166.8	256.5	668.0
	N80	2.0	235.4	364.4	867.2
	N90	3.7	319.2	496.0	1154.9
	N100	4.7	424.2	655.5	1533.6
	平均	1.4	140.4	216.2	529.8
C20	N10	0.0	1.6	0.9	2.0
	N20	0.1	8.4	9.8	26.1
	N30	0.1	21.5	31.4	106.4
	N40	0.3	42.5	63.8	192.6
	N50	0.6	71.8	109.7	309.5
	N60	0.9	112.0	173.4	473.0
	N70	1.4	166.3	257.9	666.6
	N80	2.6	231.6	366.0	881.6
	N90	3.1	315.7	497.1	1186.4
	N100	5.2	417.9	658.6	1574.2
	平均	1.4	138.9	216.9	541.8
C30	N10	0.0	1.6	0.5	1.9
	N20	0.1	8.4	7.7	30.9
	N30	0.1	20.8	31.5	99.1
	N40	0.3	42.2	64.3	189.4
	N50	0.7	71.5	110.5	295.9
	N60	1.1	111.2	175.1	457.1
	N70	1.8	164.5	259.4	668.0
	N80	2.7	230.5	367.3	867.2
	N90	3.8	314.0	502.9	1154.9
	N100	5.6	414.9	664.3	1533.6
	平均	1.6	137.9	218.3	529.8
平均	1.5	139.1	217.1	533.8	

両方を含む。

双対上昇法は、C10 問題では 2.77 ~ 9.46% で平均 7.44% の誤差であるが、C20 問題では 8.03 ~ 16.38% で平均 13.98%、C30 問題では 13.07 ~ 22.72% で平均 19.83% となり、デザイン費用が高くなるにつれて誤差が大幅に増加し、C30 問題の多くでは誤差が平均 20% を超えている。UAFB に対する Lagrange 緩和法は、C10 問題では 0.50 ~ 2.75% で

表 4.5 上界値の誤差の比較

問題		双対上昇 (%)	<i>UAFB</i> (%)	<i>UAFS</i> (%)	<i>UAFE</i> (%)
C10	N10	0.00	0.00	0.00	0.00
	N20	0.10	0.00	0.00	0.00
	N30	0.36	0.00	0.04	0.00
	N40	0.14	0.00	0.02	0.00
	N50	0.41	0.23	0.48	0.23
	N60	0.64	0.62	0.78	0.62
	N70	1.47	1.55	1.92	1.55
	N80	2.31	2.41	2.46	2.41
	N90	2.51	2.34	2.34	2.34
	N100	1.83	2.01	2.29	2.01
	平均	0.98	0.91	1.03	0.91
C20	N10	0.00	0.00	0.00	0.00
	N20	0.08	0.00	0.00	0.00
	N30	0.37	0.00	0.04	0.00
	N40	0.07	0.00	0.07	0.00
	N50	1.09	1.03	1.19	1.03
	N60	1.72	1.68	1.94	1.68
	N70	2.30	2.10	2.43	2.10
	N80	3.07	3.00	3.38	3.00
	N90	3.22	3.16	3.20	3.16
	N100	2.10	1.82	2.09	1.82
	平均	1.40	1.28	1.43	1.28
C30	N10	0.00	0.00	0.00	0.00
	N20	0.06	0.00	0.00	0.00
	N30	0.00	0.00	0.00	0.00
	N40	0.73	0.73	0.73	0.73
	N50	1.70	1.45	1.53	1.45
	N60	1.84	1.95	1.96	1.95
	N70	2.20	2.36	2.47	2.36
	N80	2.80	2.87	2.99	2.87
	N90	3.37	3.06	3.12	3.06
	N100	2.99	3.01	3.33	3.01
	平均	1.57	1.54	1.61	1.54
平均	1.32	1.24	1.36	1.24	

平均 1.80% の誤差であるが、C20 問題では 2.10~5.65% で平均 4.19%、C30 問題では 4.26~8.55% で平均 6.87% となり、デザイン費用が高くなるにつれて誤差が増加している。しかし、双対上昇法に比べて、*UAFB* の誤差は C10 問題では平均 5.64%、C20 問題では 9.79%、C30 問題では 12.96% 小さい。*UAFB* は、双対上昇法よりは良い下界値を算出しているが、デザイン費用が高い問題では平均誤差が大きい。*UAFS* に対する

表 4.6 上界値の計算時間の比較

問題		双対上昇 (秒)	<i>UAFB</i> (秒)	<i>UAFS</i> (秒)	<i>UAFE</i> (秒)
C10	N10	0.0	8.0	5.7	6.6
	N20	3.0	98.5	98.6	119.9
	N30	29.1	329.5	321.6	531.4
	N40	209.5	630.7	731.4	1172.0
	N50	779.8	1366.0	1457.9	2277.8
	N60	3108.2	2621.7	2988.1	4671.6
	N70	6818.5	3716.5	4198.3	6358.3
	N80	15502.5	5687.4	6627.0	10165.8
	N90	30848.9	8397.6	9688.0	14817.6
	N100	74953.4	12005.3	13710.9	14104.3
	平均	13225.3	3486.1	3982.7	5522.5
C20	N10	0.1	14.0	5.2	6.2
	N20	7.8	96.0	75.9	103.2
	N30	77.3	311.6	299.3	504.4
	N40	542.0	708.5	764.1	1181.1
	N50	1759.0	1266.2	1561.5	2412.8
	N60	5423.2	2264.6	2574.6	4085.5
	N70	6057.1	3801.0	4143.1	6457.2
	N80	10427.7	5846.9	6699.1	10313.4
	N90	32701.5	8681.2	9710.1	14798.9
	N100	46054.7	12406.5	13768.4	15440.3
	平均	10305.0	3539.6	3960.1	5530.3
C30	N10	0.1	15.0	2.9	5.1
	N20	2.1	106.6	56.8	83.6
	N30	29.3	318.8	298.5	502.7
	N40	160.6	631.9	730.6	1207.9
	N50	612.3	1420.9	1457.1	2318.9
	N60	2217.7	2288.9	2584.4	4081.6
	N70	4045.2	3830.5	4110.1	6421.2
	N80	10951.5	5922.0	6615.7	10190.6
	N90	19139.5	8811.8	9634.4	14892.8
	N100	49109.6	12547.6	12749.6	15318.2
	平均	8626.8	3589.4	3824.0	5502.3
平均	10719.0	3538.4	3922.3	5518.4	

Lagrange 緩和法は, C10 問題では 0.16 ~ 2.44% で平均 1.44% の誤差, C20 問題では 0.27 ~ 3.40% で平均 1.89%, C30 問題では 0.06 ~ 3.63% で平均 1.85% である. *UAFB* に対する解法に比べると, 誤差は C10 問題では平均 0.36%, C20 問題では 2.30%, C30 問題では 5.02% 小さい. 一方, *UAFE* に対する Lagrange 緩和法は, C10 問題では 0.00 ~ 2.26% で平均 1.21% の誤差であり, C20 問題では 0.03 ~ 3.18% で平均 1.60%, C30 問題

では 0.00~3.38% で平均 1.57% である。UAFB に対する解法に比べて、C10 問題では平均で 0.59%、C20 問題では 2.59%、C30 問題では 5.30% 小さく、UAFS に対する解法と比べて C10 問題では平均で 0.23%、C20 問題で 0.29%、C30 問題では 0.28% 小さく、すべての費用比において他の解法より精度が高い下界値を算出している。全体としては、UAFE に対する Lagrange 緩和法が最も良く、続いて UAFS に対する Lagrange 緩和法、UAFS に対する Lagrange 緩和法である。また、双対上昇法は、すべての問題において良い下界値を求めることができていない。

表 4.4 に、平均計算時間を示す。表 4.4 の計算時間には Lagrange/双対ヒューリスティックによる上界値の計算時間を含んでいない。双対上昇法は、C10 問題では 0.0~4.7 秒で平均 1.4 秒、C20 問題では 0.0~5.2 秒で平均 1.4 秒、C30 問題では 0.0~5.6 秒で平均 1.6 秒となり、いずれの問題に対しても高速であり、ノード数 100 の問題でも 6 秒以内の計算時間である。UAFB に対する Lagrange 緩和法は、C10 問題では 0.9~424.2 秒で平均 140.4 秒、1.6~417.9 秒で平均 138.9 秒、C30 問題では 1.6~414.9 秒で平均 137.9 秒となり、問題による差は見られないが、ノード数の増加とともに計算時間が大幅に増加している。UAFS に対する Lagrange 緩和法は、C10 問題では 1.0~655.5 秒で平均 216.2 秒、C20 問題では 0.9~658.6 秒で平均 216.9 秒、C30 問題では 0.5~664.3 秒で平均 218.3 秒となり、問題による差は見られないが、ノード数の増加とともに計算時間が大幅に増加し、UAFB に対して 1.5 倍程度の計算時間である。UAFE に対する Lagrange 緩和法は、C10 問題では 1.9~1533.6 秒で平均 529.8 秒、C20 問題では 2.0~1574.2 秒で平均 541.8 秒、C30 問題では 1.9~1533.6 秒で平均 529.8 秒となり、問題による差は見られないが、ノード数の増加とともに計算時間が大幅に増加し、UAFS に対して 2.5 倍程度の計算時間である。これは、Lagrange 乗数の数が多く、解の収束が遅くなることから、他の緩和法の 1.5 倍の繰返し回数を行っていることが要因である。

全体としては、双対上昇法は非常に高速であり、UAFE に対する Lagrange 緩和法が最も計算時間を要している。これは、これらの問題では双対上昇法の繰返しは数回であり、Lagrange 緩和法では劣勾配法の繰返しを最大で 10000~15000 回行っているためである。計算時間は、劣勾配法の繰返し回数に比例するため、繰返し回数を減らせば計算時間は減少するが、解の精度は悪化する。

表 4.5 に、従来の解法であるバックワード法と三種類の Lagrange 緩和法による上界値の平均誤差を示す。ノード数 30 までの下界値は最適値であるため、平均誤差は最適値と上界値との差である。また、C10 問題と C20 問題はノード数 40 まで、C30 問題はノード数 30 までは最適値であるため、それら以外の誤差には、上界値と最適値の差、および最適値と下界値との差の両方を含んでいる。

双対上昇法は、C10 問題では 0.00~2.51% で平均 0.98% の誤差、C20 問題では 0.00~3.22% で平均 1.40%、C30 問題では 0.00~3.37% で平均 1.57% となり、費用比による大きな差は見られない。UAFB に対する Lagrange 緩和法は、C10 問題では 0.00~2.41% で平均 0.91% の誤差、C20 問題では 0.00~3.16% で平均 1.28%、C30 問題では 0.0~3.06% で平均 1.54% となり、費用比による大きな差は見られない。UAFS に対す

る Lagrange 緩和法は，C10 問題では 0.00～2.46% で平均 1.03% の誤差，C20 問題では 0.00～3.38% で平均 1.43%，C30 問題では 0.00～3.33% で平均 1.61% となり，費用比による大きな差は見られない．また，*UAFB* に対する Lagrange 緩和法よりも悪いものも見受けられる．*UAFE* に対する Lagrange 緩和法は，C10 問題では 0.00～2.41% で平均 0.91% の誤差，C20 問題では 0.00～3.16% で平均 1.28%，C30 問題では 0.00～3.06% で平均 1.54% となり，費用比による大きな差は見られない．

双対上昇法による下界値の誤差は大きい，比較的良い上界値が求まっている．これは，双対上昇法における双対ヒューリスティックに，双対解をもとにした限定しないフォワード法，限定しないバックワード法，アーク交換による局所探索法を行っているためであるが，計算時間は大きなものとなる．また，三種類の Lagrange 緩和法は同じ Lagrange ヒューリスティックを行っているため，ほぼ同じ結果となっている．また，第3章で示した Minoux 法の改良法と比べると，C10 問題では 0.43～0.55%，C20 問題では 0.57～0.72%，C30 問題では 0.63～0.70% 良い解が求められている．また，線形緩和法と比べると，C10 問題では 0.04～0.16% 劣り，C20 問題では同値か 0.15% 劣り，C30 問題では 0.02～0.09% 劣っている．

表 4.6 に，Lagrange ヒューリスティックおよび双対ヒューリスティックを含めた平均計算時間を示す．双対上昇法は，C10 問題では 0.0～74953.4 秒で平均 13225.3 秒，C20 問題では 0.1～46054.7 秒で平均 10305.0 秒，C30 問題では 0.1～49109.6 秒で平均 8626.8 秒となり，小さな問題では計算時間は短い，ノード数の増加と共に計算時間が増大し，ノード 60～80 以上の問題では Lagrange 緩和法よりも計算時間が大幅に増加している．これは，計算量の大きなアーク交換による限定しない局所探索法を行っているためである．*UAFB* に対する Lagrange 緩和法は，C10 問題では 8.0～12005.3 秒で平均 3486.1 秒，C20 問題では 14.0～12406.5 秒で平均 3539.6 秒，C30 問題では 15.0～12547.6 秒で平均 3589.4 秒となり，費用比による差は見られないが，ノード数の増加とともに計算時間が大幅に増加している．*UAFS* に対する Lagrange 緩和法は，C10 問題では 5.7～13710.9 秒で平均 3982.7 秒，C20 問題では 5.2～13768.4 秒で平均 3960.1 秒，C30 問題では 2.9～12749.6 秒で平均 3824.0 秒となり，費用比による差は見られないが，ノード数の増加とともに計算時間が大幅に増加している．*UAFE* に対する Lagrange 緩和法は，C10 問題では 6.6～14104.3 秒で平均 5522.5 秒，C20 問題では 6.2～15440.3 秒で平均 5530.3 秒，C30 問題では 5.1～15318.2 秒で平均 5502.3 秒となり，費用比による差は見られないが，ノード数の増加とともに計算時間が大幅に増加している．表 4.4 と表 4.6 の差が，Lagrange ヒューリスティックに要した計算時間であり，下界値の算出よりも，上界値の算出に要する時間が非常に大きい．

4.8 本章のまとめ

本章では，容量制約のないネットワーク設計問題に対して，アークフローによる定式化に対する三種類の Lagrange 緩和法を示した．これは，フロー保存式といくつかの強い妥

当不等式を Lagrange 緩和した問題を用いた方法である。第一の解法は、非順流不等式を含まない弱い定式化に対する方法である。第二の解法は、一種類の非順流不等式を含む定式化に対する方法である。第三の解法は、二種類の非逆流不等式と二種類の非順流を含む強い定式化に対する方法である。また、数値実験を行い、提案した解法の有効性を示した。

本章では、容量制約のないネットワークデザイン問題に対して、以下のような成果を示した。

1) 三種類の定式化を示した。

弱い定式化は、品種と需要に関する強制制約を含む定式化である。強い定式化は、始点を同じにする品種のフローに対する一種類の非逆流不等式とネットワークの連結条件を含む定式化である。拡張した定式化は、二種類の非逆流不等式、二種類の非順流不等式とネットワークの連結条件を含む定式化である。

2) 弱い定式化 $UAFB$ に対する Lagrange 緩和法を示した。

フロー保存式を Lagrange 緩和した緩和問題を示し、この Lagrange 緩和問題がアーク毎の問題に分割できることを示した。さらに、分割された問題が係数の比較により容易に解けることを示し、緩和問題がデザイン変数のみを含む問題として表現できることを示した。また、Lagrange 乗数の変更法として、勾配法を示した。

3) 強い定式化 $UAFS$ に対する Lagrange 緩和法を示した。

フロー保存式を Lagrange 緩和し、非逆流不等式を制約式とした緩和問題を示し、この Lagrange 緩和問題がアーク毎、始点毎の問題に分割することができることを示した。さらに、分割された問題が最小木問題に帰着されることを示し、緩和問題がデザイン変数のみを含む問題として表現できることを示した。また、Lagrange 乗数の変更法として、勾配法を示した。

4) 拡張した定式化 $UAFE$ に対する Lagrange 緩和法を示した。

フロー保存式、二種類の非逆流のうち一種類と、二種類の非順流不等式を Lagrange 緩和し、一種類の非逆流不等式を制約式として残した Lagrange 緩和問題を示し、この Lagrange 緩和問題がアーク毎、始点毎の問題に分割することができることを示した。さらに、Lagrange 緩和問題が最小木問題に帰着されることを示し、緩和問題がデザイン変数のみを含む問題として表現できることを示した。また、Lagrange 乗数の変更法として、勾配法を示した。Lagrange 乗数が非常に多くなるため、必要に応じて Lagrange 乗数の値を設定する方法を示した。

5) Lagrange 緩和解をもとにした三種類の Lagrange ヒューリスティックを示した。

Lagrange 緩和解において 1 であるデザイン変数と、係数の小さな一定このデザイン変数を用いた解を初期解とし、この初期解をもとにした限定した Minoux タイプのフォワード法、限定した Minoux 法の改良法、限定した局所探索法を示した。

6) 数値実験を行い、従来の解法と比較し、提案した解法の有効性を示した。

下界値に対して、従来の解法である双対上昇法と比較した数値実験を行った。いずれの解法も双対上昇法よりも優れた下界値を算出することができた。特に、

$UAFE$ は最も良い上界値と下界値を算出することができた。また、「デザイン費用 / フロー費用」が大きくなると $UAFB$ では誤差が大きくなるが、 $UAFS$ および $UAFE$ では安定して良い下界値を算出することができた。また、第3章で示した近似解法と比べて、Minoux の改良法に対しては計算時間はかかるものの規模の大きな問題では良い解を求めることができ、線形緩和法と同程度の解を求めることができた。

7) 問題の属性と適用すべき解法

きわめて短時間で下界値を求めたい場合は双対上昇法を適用し、ある程度短時間で下界値を求めたい場合は提案した $UAFS$ を適用することが良いと考える。また、精度の高い下界値を求めたい場合は提案した $UAFE$ を適用し、精度の高い解を算出する近似解を求めたい場合にはいずれの解法を適用しても良いと考える。

今後の課題としては、次の項目が挙げられる。

1) 大規模な問題に対する誤差の低減

$UAFS$, $UAFE$ では、ノード数が 70 を超えると誤差が 2% を超える。このため、規模の大きな問題における誤差の改善が課題であり、新たな妥当不等式の考慮などが必要となる。

2) 大規模な問題に対する計算時間の短縮

ノード数が増加すると計算時間が膨大なものとなる。特に、Lagrange ヒューリスティックの計算時間が大きくなっている。このため、解の精度を大きく変えない範囲で、Lagrange ヒューリスティックの実施回数の適切な制御や Lagrange ヒューリスティック自体の高速化などにより、計算時間を短縮することが必要である。

第5章

容量制約をもつネットワーク設計問題の近似解法

5.1 本章の概要

本章では、容量制約をもつネットワーク設計問題 (CND) を対象とする。 CND は、アーク上を移動するフロー量がアーク容量以下であるという容量制約をもつ問題である。アーク容量は輸送能力や輸送便数に相当し、これらを考慮しているため、 UND に比べより現実的なモデルとなる。しかし、容量制約があるために、 CND は UND や BND に比べて最適値と緩和問題による下界値と差であるギャップが大きくなる傾向がある。また、デザイン変数を固定したネットワーク上でフローを求める問題が多品種フロー問題となるため、実行可能解を求めること自体に手間がかかる。さらに、貪欲法、Lagrange 緩和や Lagrange ヒューリスティックが有効に機能しないなど、 UND や BND よりも難しい問題となる。

CND に対しては、メタ解法とよばれる近似解法が数多く開発されている。しかし、容量制約のない問題である UND や BND に比べて、規模のそれほど大きくない問題であっても、高速に良い近似解を算出する解法が開発されていない状況である。そのため、 CND に対して、従来の研究よりも平均的に精度が高く、かつ安価なパソコンを用いても高速に解くことができる解法が必要となる。

本章では、 CND に対する二種類の容量スケールリング法を用いた近似解法を提案する。容量スケールリング法は、線形緩和問題の解をもとに、アーク容量を変更しながら、最適に近いフローを探索していく方法である。第一の解法はアークフロー変数を用いた定式化にもとづく方法であり、第二の解法はパスフロー変数を用いた定式化にもとづく方法である。

はじめに、 CND の前提条件、記号および CND の定義、アークフローを用いた定式化とパスフローを用いた定式化を示す。続いて、アークフロー変数を用いた定式化にもとづく容量スケールリング法とパスフロー変数を用いた定式化にもとづく容量スケールリング法を示す。さらに、提案した解法の有効性を示すために数値実験を行い、その結果を示す。最

後に、本章のまとめを記述する。

5.2 問題の定式化

はじめに、 CND の前提条件、使用する記号および CND の定義を示す。続いて、アークフローを用いた定式化とパスフローを用いた定式化を示す。

5.2.1 前提条件、記号および問題の定義

前提条件

- ノード集合が与えられている。
- アーク集合が与えられている。
- アークは向きをもつ。
- アークには、非負のデザイン費用が与えられている。
- アークには、単位当たりの非負のフロー費用が与えられている。
- アークには、単位期間当たりの処理量の上限であるアーク容量が与えられている。
- 複数の品種からなる品種集合が与えられている。
- 各品種の需要が与えられている。
- 各品種の需要は、始点から終点までのパス上を移動する。

記号の定義

- N : ノード集合
- A : アーク集合
- K : 品種集合
- N_n^+ : ノード n を始点とするアークの終点であるノード集合
- N_n^- : ノード n を終点とするアークの始点であるノード集合
- P^k : 品種 k の取り得るパス集合
- \bar{P}^k : 品種 k の取り得るパス集合の部分集合
- \bar{P} : P の部分集合である \bar{P}^k の和集合
- $A_{\bar{P}^k}$: \bar{P}^k に含まれるパスに含まれるアーク集合
- c_{ij}^k : アーク (i, j) 上における品種 k の非負のフロー費用
- f_{ij} : アーク (i, j) の非負のデザイン費用
- C_{ij} : アーク (i, j) のアーク容量
- C_{ij}^l : l 回目の繰返しにおけるアーク (i, j) のアーク容量
- d^k : 品種 k の需要
- δ_{ij}^p : パス p にアーク (i, j) が含まれるとき 1, そうでないとき 0 を表す定数
- O^k : 品種 k の始点
- D^k : 品種 k の終点

- x_{ij}^k : 品種 k のフローがアーク (i, j) 上を移動する量を表すアークフロー変数 ; 非負の連続変数
- X_{ij} : アーク (i, j) 上を移動する量を表す総フロー変数 ; 非負の連続変数
- z_p^k : 品種 k のフローがパス p 上を移動する量を表すパスフロー変数 ; 非負の連続変数
- y_{ij} : アーク (i, j) を選択するとき 1 , そうでないとき 0 であるデザイン変数 ; 0-1 変数
- π^k : 品種 k に関する需要保存式に対する双対変数 : 連続変数
- u_{ij} : アーク (i, j) に関する容量制約式に対する双対変数 : 非負の連続変数
- w_{ij}^k : アーク (i, j) における品種 k の需要に関する強制制約式に対する双対変数 : 非負の連続変数
- g_{ij} : アーク (i, j) のデザイン変数の上限制約式に対する双対変数 : 非負の連続変数

問題の定義

ノード集合 N , デザイン費用 f , フロー費用 c , アーク容量 C をもつ向きをもつアーク集合 A , 品種の需要 d をもつ品種集合 K が与えられている . このとき , フロー費用とデザイン費用の合計を最小にするアーク集合 $A' (\subseteq A)$, およびアーク容量を満足するフロー x または z を求めよ .

5.2.2 アークフローを用いた定式化

アークにおける品種の需要に関する強制制約式を含まない CND のアークフローによる弱い定式化を $CAFW$ とする . $CAFW$ は , 次のようになる .

($CAFW$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.1)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.2)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.3)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.4)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.5)$$

(5.1) 式は目的関数であり , フロー費用とデザイン費用の総和を最小化する . (5.2) 式はフロー保存式であり , ノードに流入するフローと流出するフローの差が , 品種 k の始点であれば $-d^k$, 終点であれば d^k , その他のノードであれば 0 であることを表す . この式は , 各品種について , 必ず始点から終点まで需要が移動することを保証する . (5.3) 式は , 容量

制約式である．この式は，アーク (i, j) が選択されるときはアーク上を移動するフロー量の合計がアーク容量以下であり，アークが選択されないときは0であることを表す．(5.4)式はアークフロー変数の非負条件，(5.5)式はデザイン変数の0-1条件である．

$CAFW$ は， $|A||K|$ 個のアークフロー変数， $|A|$ 個のデザイン変数，および $|N||K| + |A|$ 本の制約式をもつ問題となる．品種の需要に関する強制制約式を含まない定式化であるため，下界値が劣る弱い定式化となる．

アークにおける品種の需要に関する強制制約式を含む CND のアークフローによる強い定式化を $CAFS$ とする． $CAFS$ は，次のようになる．

($CAFS$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.6)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.7)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.8)$$

$$0 \leq x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5.9)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.10)$$

(5.9)式は，アーク (i, j) における品種 k の需要 d^k に関する強制制約式である．この式は，アーク (i, j) が選択されるときには品種 k のフローが最大で d^k となり，アーク (i, j) が選択されないときには0となることを表す． $d^k > C_{ij}$ である品種の需要が存在する場合，この制約式の右辺は $\min(d^k, C_{ij})y_{ij}$ に置き換えることができる．

$CAFS$ は， $|A||K|$ 個のアークフロー変数， $|A|$ 個のデザイン変数，および $|N||K| + |A| + |A||K|$ 本の制約式をもつ問題となる． $CAFS$ は強制制約式を含む定式化であるため， $CAFW$ よりも強い定式化となり， $CAFS$ の線形緩和による下界値は $CAFW$ による下界値よりも優れている．しかし，(5.9)式があるため，制約式の数が増加し， $CAFW$ よりも難しい問題となる．

5.2.3 パスフローを用いた定式化

CND のパスフローによる強い定式化 $CPFS$ を示す．

(CPFS)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.11)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (5.12)$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p z_p^k \leq C_{ij} y_{ij} \quad \forall (i,j) \in A \quad (5.13)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall k \in K, (i,j) \in A \quad (5.14)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (5.15)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \quad (5.16)$$

(5.11) 式は目的関数であり，フロー費用とデザイン費用の総和を最小化する．ここで， $\sum_{p \in P^k} \delta_{ij}^p z_p^k$ は x_{ij}^k に一致する．(5.12) 式は，品種 k のパスフローの合計が品種 k の需要 d^k に一致することを表す需要保存式である．(5.13) 式は，アーク (i, j) が選択されるときはアーク上を移動するフロー量の合計がアーク容量以下であり，アークが選択されないときは 0 であることを表す容量制約式である．(5.14) 式は，アーク (i, j) における品種 k の需要 d^k に関する強制制約式である．(5.15) 式はパスフローの変数の非負制約であり，(5.16) 式はデザイン変数の 0-1 条件である．

CPFS は， $\sum_{k \in K} |P^k|$ 個である指数個のパスフロー変数， $|A|$ 個のデザイン変数と $|K| + |A| + |A||K|$ 本の制約式をもつ問題となる．変数が指数個と膨大なものとなるので，小規模な問題であってもこの定式化を直接解くことは困難である．実際には，逐次，必要なパスフロー変数を生成して問題を解く列生成法が用いられる．この列生成法をうまく適用すれば，アークフローによる定式化の場合よりも，陽的に使用する変数の数を抑えることができる．

5.2.4 CAFW と CAFS に対する緩和問題

線形緩和問題

CAFW に対して，デザイン変数の 0-1 条件を非負条件に線形緩和した問題 CAFWL¹ を考える．

(CAFWL¹)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.17)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.18)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.19)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.20)$$

$$y_{ij} \geq 0 \quad \forall (i, j) \in A \quad (5.21)$$

この問題は最小化問題であり, f_{ij} は非負, y_{ij} の下限は (5.19) 式のみである. したがって, 最適解において (5.19) 式は等式で成り立つため $y_{ij} = \sum_{k \in K} x_{ij}^k / C_{ij}$ となる. これを目的関数に代入すると, CAFWL¹ はデザイン変数を含まない次のような線形化フロー問題 LZF に帰着される.

(LZF)

$$\text{最小化} \quad \sum_{k \in K} \sum_{(i,j) \in A} \left(c_{ij}^k + \frac{f_{ij}}{C_{ij}} \right) x_{ij}^k \quad (5.22)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.23)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.24)$$

LZF は品種 k 毎の独立した問題に分解でき, それぞれの問題はアーク (i, j) の長さを $c_{ij}^k + f_{ij}/C_{ij}$ とした品種 k の始点・終点間の最短路問題に帰着され, 容易に解くことができる.

次に, 問題 CAFW において, デザイン変数の 0-1 条件を 0 から 1 の連続数に緩和した線形問題 CAFWL² を考える. CAFWL² は次のようになる.

(CAFW²)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.25)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.26)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.27)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.28)$$

$$0 \leq y_{ij} \leq 1 \quad \forall (i, j) \in A \quad (5.29)$$

$CAFWL'$ と同様に, $y_{ij} = \sum_{k \in K} x_{ij}^k / C_{ij}$ を用いると, $CAFWL^2$ は次のような問題 MCF に帰着される.

(MCF)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \left(c_{ij}^k + \frac{f_{ij}}{C_{ij}} \right) x_{ij}^k \quad (5.30)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.31)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} \quad \forall (i, j) \in A \quad (5.32)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.33)$$

$$(5.34)$$

MCF は, 容量制約をもつネットワーク上で多品種のフローを求める問題, すなわち多品種フロー問題となる.

$CAFS$ に対して, デザイン変数の 0–1 条件を 0 から 1 の連続数に緩和した線形問題 $CAFSL$ を考える. $CAFSL$ は次のようになる.

($CAFSL$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.35)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.36)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.37)$$

$$0 \leq x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5.38)$$

$$0 \leq y_{ij} \leq 1 \quad \forall (i, j) \in A \quad (5.39)$$

$CAFSL$ には y の下限の制約として (5.38) 式が加わるため, 最適解において (5.37) 式が等号で成り立つとは限らない. このため, $CAFSL$ は y を含まない問題として表すことはできない. $CAFSL$ は線形計画問題ではあるが, 多くの制約式を含むため規模が大きな場合はそれほど容易に解ける訳ではない.

$CAFWL^1$ と $CAFWL^2$ は, 最短路問題や多品種フロー問題の解法を用いて解くことができる. 一方, $CAFSL$ は線形計画問題ではあるが, 問題の規模が大きくなると容易に解くことはできない. $CAFWL^1$ と $CAFWL^2$ は強制制約式 (5.9) を考慮していないため, $CAFSL$ に比べてこれらの線形緩和問題から得られる下界値は良いものではなく, CND の最適値と下界値の差であるギャップが大きくなる傾向がある.

Lagrange 緩和問題

$CAFS$ において, Lagrange 乗数 v を用いて, フロー保存式 (5.7) を Lagrange 緩和した問題 $CAFK$ を作成する. $CAFK$ は次のようになる.

($CAFK$)

$$\text{最小化} \quad \sum_{k \in K} d^k (v_{D^k}^k - v_{O^k}^k) + \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.40)$$

$$\text{条件} \quad \sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.41)$$

$$0 \leq x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5.42)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.43)$$

適当な v が与えられたときに目的関数の第一項 $\sum_{k \in K} d^k (v_{D^k}^k - v_{O^k}^k)$ は定数項となるので, $CAFK$ はアーク (i, j) 毎の独立した次のような問題 $CAFK_{ij}$ に分割できる.

($CAFK_{ij}$)

$$\text{最小化} \quad \sum_{k \in K} (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + f_{ij} y_{ij} \quad (5.44)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad (5.45)$$

$$0 \leq x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K \quad (5.46)$$

$$y_{ij} \in \{0, 1\} \quad (5.47)$$

$CAFK_{ij}$ に含まれる 0-1 変数は y_{ij} のみであるので, $y_{ij} = 0$ と $y_{ij} = 1$ の場合に分けて考える.

$y_{ij} = 1$ の場合, $CAFK_{ij}$ は次のような問題 $CAFK_{ij}^1$ になる.

($CAFK_{ij}^1$)

$$\text{最小化} \quad \sum_{k \in K} (c_{ij}^k - v_j^k + v_i^k) x_{ij}^k + f_{ij} \quad (5.48)$$

$$\text{条件} \quad \sum_{k \in K} x_{ij}^k \leq C_{ij} \quad (5.49)$$

$$0 \leq x_{ij}^k \leq d^k \quad \forall k \in K \quad (5.50)$$

この問題は連続ナップサック問題となり, 目的関数における変数の係数をソートすることにより容易に解くことができる.

$(c_{ij}^k - v_j^k + v_i^k)$ を昇順にソートし, 昇順に l 番目の品種を h_l とおく. $\sum_{l=1}^{m-1} d^{h_l} \leq C_{ij}$ かつ $\sum_{l=1}^m d^{h_l} > C_{ij}$ を満足する m を用いると, $CAFK_{ij}^1$ の最適解 \tilde{x} は次のようになる.

$$\tilde{x}_{ij}^k = \begin{cases} d^k & \text{if } l < m \\ C_{ij} - \sum_{l=1}^{m-1} d^{h_l} & \text{if } l = m \\ 0 & \text{if } l > m \end{cases} \quad \forall k \in K, (i, j) \in A$$

一方, $y_{ij} = 0$ の場合, $CAFK_{ij}$ の最適解は $x_{ij} = 0$, 最適値は 0 となる. 最小化問題であるため, $CAFK_{ij}$ の最適解 \hat{y}, \hat{x} は, $CAFK_{ij}^1$ の最適値が非負であれば $\hat{y}_{ij} = 0$ および $\hat{x}_{ij} = 0$, 負であれば $\hat{y}_{ij} = 1$ および \hat{x}_{ij} となる.

次に, 容量制約式 (5.8) と強制制約式 (5.9) の右側の式に対する非負の Lagrange 乗数 u と w を用いて, Lagrange 緩和問題 $CAFF$ を作成する. $CAFF$ は次のようになる.

($CAFF$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + u_{ij} + w_{ij}^k) x_{ij}^k + \sum_{(i,j) \in A} \left(f_{ij} - C u_{ij} - \sum_{k \in K} d^k w_{ij}^k \right) y_{ij} \quad (5.51)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.52)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.53)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.54)$$

$CAFF$ は, 次のような x に関する問題 $CAFFx$ と y に関する問題 $CAFFy$ に分割できる.

($CAFFx$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + u_{ij} + w_{ij}^k) x_{ij}^k \quad (5.55)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.56)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.57)$$

($CAFFy$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \left(f_{ij} - C u_{ij} - \sum_{k \in K} d^k w_{ij}^k \right) y_{ij} \quad (5.58)$$

$$\text{条件} \quad y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.59)$$

$CAFFx$ は品種毎の始点・終点間の最短路問題に分割することができ, 容易に解くことができる. また, $CAFFy$ は y の目的関数の係数の正負により解くことができる.

5.2.5 CPFS に対する被約費用と双対問題

パスフロー変数の被約費用

パスフローによる定式化 $CPFS$ において, パスフロー変数 z に対する被約費用を求める. (5.12) 式に対する双対変数 π , (5.13) および (5.14) 式に対する非負の双対変数 u, w を用いて, z に関する Lagrange 双対関数 $LF(z)$ を作成する.

$$\begin{aligned}
LF(\mathbf{z}) &= \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p z_p^k - \sum_{k \in K} \sum_{p \in P^k} z_p^k \pi^k \\
&\quad + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p z_p^k u_{ij} + \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p z_p^k w_{ij}^k \\
&= \sum_{k \in K} \sum_{p \in P^k} \left\{ \sum_{(i,j) \in A} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k) - \pi^k \right\} z_p^k \quad (5.60)
\end{aligned}$$

したがって、パスフロー変数 z に関する被約費用は次のようになる。

$$\sum_{(i,j) \in A} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k) - \pi^k \quad \forall p \in P^k, k \in K \quad (5.61)$$

(5.61) 式の第一項はアークの長さを $c_{ij}^k + u_{ij} + w_{ij}^k$ としたパス p の長さであり、 π^k は品種 k の始点・終点間の最短距離に相当する。

双対問題

$CPFS$ に対して、デザイン変数の 0-1 条件 (5.16) 式を 0 から 1 の連続数に緩和した線形問題 $CPFSL$ を考え、 $CPFSL$ に対する双対問題を示す。(5.12) 式に対する双対変数を π 、(5.13) および (5.14) 式に対する非負の双対変数を u 、 w とし、デザイン変数の上限式に対する非負の双対変数を g とする。このとき、 $CPFSL$ に対する双対問題 DU は次のようになる。

(DU)

$$\text{最大化} \quad \sum_{k \in K} d^k \pi^k - \sum_{(i,j) \in A} g_{ij}^s \quad (5.62)$$

$$\text{条件} \quad \sum_{(i,j) \in A} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k) \geq \pi^k \quad \forall p \in P^k, k \in K \quad (5.63)$$

$$c_{ij} u_{ij} + \sum_{k \in K} d^k w_{ij}^k - g_{ij} \leq f_{ij} \quad \forall (i,j) \in A \quad (5.64)$$

$$u_{ij} \geq 0 \quad \forall (i,j) \in A \quad (5.65)$$

$$w_{ij}^k \geq 0 \quad \forall k \in K, (i,j) \in A \quad (5.66)$$

$$g_{ij} \geq 0 \quad \forall (i,j) \in A \quad (5.67)$$

(5.63) 式の左辺は、アークの長さを $c_{ij}^k + u_{ij} + w_{ij}^k$ としたパス p の長さである。最大化問題であることから、 π^k は品種 k のパスの長さの最小値、すなわち品種 k の始点・終点間の最短距離に相当する。

5.3 アークフローを用いた定式化に対する容量スケールリング法

容量スケールリング法は、線形緩和問題を解くことによって得られた解を用いてパラメータ化されたアーク容量を変化させることを繰り返し、得られたデザイン変数をもとに近似解を算出する方法である。はじめに、容量スケールリング法の内容を示し、続いてアークフローを用いた定式化に対する容量スケールリング法を示す。

5.3.1 容量スケールリング

なんらかの手段を用いて、 CND の最適なフロー量を求めることができれば、アーク容量をこれらの量に変更しても、 CND の最適値は変わらない。さらに、このようにアーク容量を変更した問題の線形緩和問題を考える。この線形緩和問題においても、最適なフロー量は実行可能解となる。緩和問題においてアーク容量を最適なフロー量に近づけると、デザイン変数が 1 に近くなることが期待できる。ただし、最適なフロー量が 0 である場合、デザイン変数は 0 と考える。しかし、最適なフロー量を求めること自体が問題の目的であるため、実際にはこれらの値を直接求めることはできない。また、アーク容量を変化させるとフロー量も変化する。

そこで、逆に、線形緩和問題において、フロー量に近づくように、少しずつアーク容量を変更しながら問題を解き直すことを繰り返す。このようにすると、アーク容量とフロー量が徐々に近づくため、デザイン変数が 0 または 1 に近づくことが期待できる。容量スケールリング法は、このように線形緩和問題の解をもとに、アーク容量を変更して繰り返し線形緩和問題を解き、0-1 変数解を導く近似解法である。

すべてが 0 または 1 であるデザイン解を求められた場合、このデザイン解をもとにフロー問題を解き、アークフローをおよび上界値を求めることができる。そうでない場合は、適当な方法を用いて、0 または 1 でないデザイン解を 0 または 1 に設定して、実行可能解を求めることが必要となる。

5.3.2 弱い定式化に対する容量スケールリング法

弱い定式化 $CAFW$ に対する容量スケールリング法を示す。はじめに、 $CAFW$ に対して、デザイン変数の 0-1 条件を 0 から 1 までの連続数に線形緩和した問題 $CAFWL^2$ を考える。

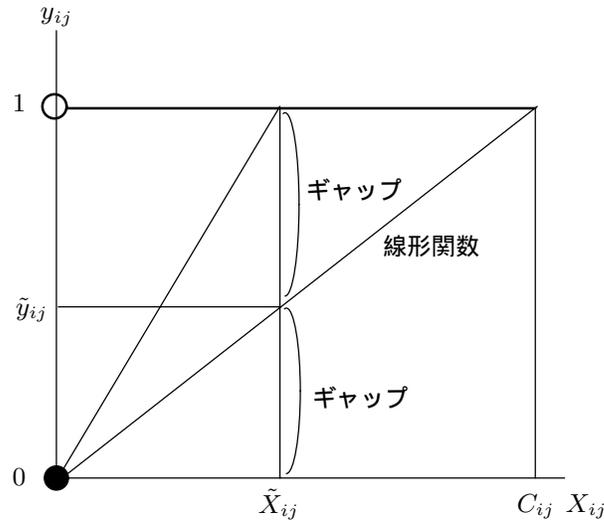


図 5.1 線形関数とギャップ

(CAFWL²)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.68)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.69)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (5.70)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.71)$$

$$0 \leq y_{ij} \leq 1 \quad \forall (i, j) \in A \quad (5.72)$$

アーク上の総フローを X_{ij} とおく．アークフローと総フローの関係式は，次式となる．

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (5.73)$$

CAFWL² は最小化問題であり， y_{ij} の下限に関する制約式は (5.70) 式のみである．フロー変数とデザイン変数の関係は，(5.70) 式から $y_{ij} = \sum_{k \in K} x_{ij}^k / C_{ij} = X_{ij} / C_{ij}$ となり，図 5.1 に示すように線形関数で表わすことができる．

CAFWL² におけるアーク (i, j) 上の総フロー \tilde{X}_{ij} が求められたとき，デザイン変数解は $\tilde{y}_{ij} = \tilde{X}_{ij} / C_{ij}$ となる． y_{ij} は本来 0 または 1 である．そのため，図 5.1 に示すように， \tilde{y}_{ij} が小数値を取った場合にはこれらの間にギャップが発生する．そこで，フローが変化しないものと想定すると，フローが存在する場合にはアーク容量 C_{ij} を \tilde{X}_{ij} に変更すれば $\tilde{y}_{ij} = 1$ となる．また，フローが存在しない場合にはアーク容量 C_{ij} を 0 に変更し， $\tilde{y}_{ij} = 0$ と考える．このようにすれば，ギャップが埋まることが期待できる．

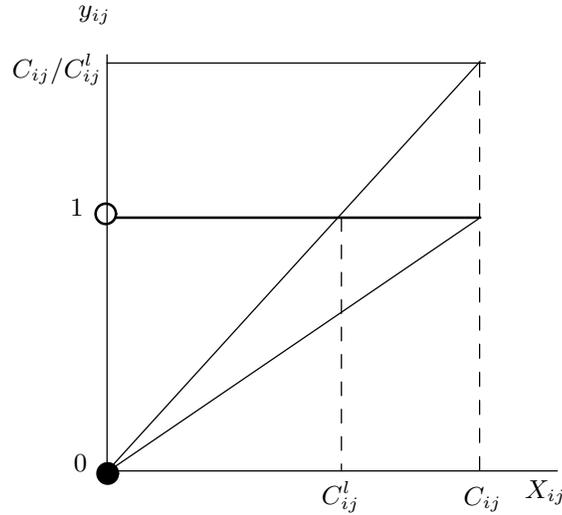


図 5.2 デザイン変数とフロー変数の関係

$CAFWL^2$ において，アーク容量を変化させることを繰り返す．繰り返し回数が l のときのアーク (i, j) の容量を C_{ij}^l とすると， $CAFWL^2$ の l 回目の繰り返しにおける $CAFWL_l$ は次のようになる．

($CAFWL_l$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.74)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = d_n^k \quad \forall n \in N, k \in K \quad (5.75)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij}^l y_{ij} \quad \forall (i, j) \in A \quad (5.76)$$

$$0 \leq y_{ij} \leq \frac{C_{ij}}{C_{ij}^l} \quad \forall (i, j) \in A \quad (5.77)$$

アーク容量を C_{ij} から C_{ij}^l に変更しているが，アーク上のフロー量は本来の容量 C_{ij} 以下であればよい．そこで，(5.72) 式の上限值である 1 を変更し， $y_{ij}^s \leq C_{ij}/C_{ij}^l$ とする．図 5.2 に示すように，このとき， y_{ij} は 1 以上の値を取ることができるが，アーク上のフロー量は本来の容量以下であることが保証される．

このように，容量スケールリング法では，変更した容量以上のフローを認める．このため， C_{ij}^l は，厳密な意味での容量ではなく， y_{ij} の取り得る範囲を変更させることにより，線形緩和におけるデザイン費用を変化させるパラメータ的な機能を果たしていることになる． $CAFWL_l$ は線形計画問題であるため，汎用の数理最適化ソフトウェアを用いて解くことができる．

実際には，アーク容量を変更させた問題を解き直すと，区分フロー自体が変化してしまい，ギャップが埋まる保証はない．そこで，パラメータ λ を用いて，アーク容量を徐々に変更した問題を解き直し，新たな解におけるフローが大きく変化ないように制御する．

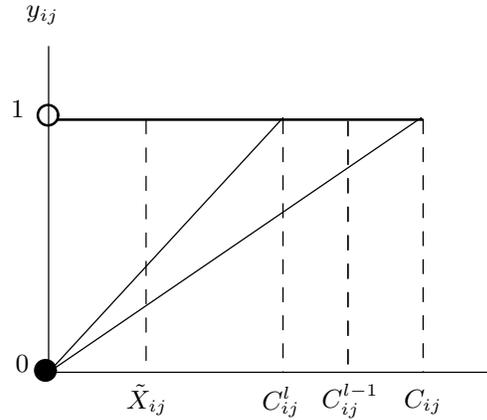


図 5.3 アーク容量の変更

デザイン変数解が 0 または 1 に収束するまで，この操作を繰り返す．

アーク容量の初期値 C^1 を C とする． $l-1$ 回目の繰り返しにおいて，容量を C^{l-1} とした $CAFWL_l$ を解き直し，アークフロー \tilde{X} を求める．続いて，パラメータ $\lambda (0 < \lambda \leq 1)$ を用いて， C^l を次のように更新する．

$$C_{ij}^l := \lambda \tilde{X}_{ij} + (1 - \lambda) C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.78)$$

図 5.3 に示すように，(5.78) 式は， l 回目のアーク容量 C_{ij}^l を現在の総フロー \tilde{X}_{ij} と $l-1$ 回目のアーク容量 C_{ij}^{l-1} の凸結合で表わし，これらの間に l 回目のアーク容量 C_{ij}^l を設定することを意味する． λ によって，アーク容量と $CAFWL_l$ を解いた結果として得られるフロー量の急激な変化を制御する．

一方， $\tilde{X}_{ij} = C_{ij}^{l-1} \tilde{y}_{ij}$ の関係から，(5.78) 式は次のようにも表現できる．

$$C_{ij}^l := \lambda C_{ij}^{l-1} \tilde{y}_{ij} + (1 - \lambda) C_{ij}^{l-1} = \{1 - \lambda(1 - \tilde{y}_{ij})\} C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.79)$$

この式は， $\tilde{y}_{ij} < 1$ の場合は，容量を $\lambda(1 - \tilde{y}_{ij})$ の比率で減少させることを意味する．また， $\tilde{y}_{ij} > 1$ の場合は， $\lambda(\tilde{y}_{ij} - 1)$ の比率で増加させることを意味する．

容量スケールリング法では，繰り返し回数とともに多くのデザイン変数は 0 または 1 に収束する．しかしながら，すべてのデザイン変数が 0 または 1 に収束する保証はないため，繰り返し回数の上限 l_{max} を設定し，一定回数の繰り返しで計算を打ち切る． $CAFWL$ に対する容量スケールリング法の流れを示す．

CAFWL に対する基本的な容量スケールリング法

- [ステップ 1] $C^1 := C$ ， $l := 0$ とする．繰り返し回数の上限を l_{max} とする．
- [ステップ 2] $l := l + 1$ とする． $CAFWL_l$ を解き，最適解 \tilde{y} ， \tilde{x} を求める．
- [ステップ 3] $C^l := \lambda \tilde{X} + (1 - \lambda) C^{l-1}$ または $C^l := \{1 - \lambda(1 - \tilde{y})\} C^{l-1}$ とする．
- [ステップ 4] \tilde{y} の成分がすべて 0 または 1 であるか， $l = l_{max}$ であれば終了，そうでなければ，ステップ 2 へ戻る．

5.3.3 強い定式化に対する容量スケールリング法

$CAFS$ に対して、デザイン変数の 0-1 条件を 0 から 1 の連続数に緩和した線形緩和問題 $CAFSL$ を作成する。 $CAFSL$ において、繰り返し毎にアーク容量を変化させる。繰り返し回数 l のときのアーク (i, j) の容量を C_{ij}^l とすると、 $CAFSL$ の l 回目の繰り返しにおける問題 $CAFSL_l$ は次のようになる。

($CAFSL_l$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.80)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = d_n^k \quad \forall n \in N, k \in K \quad (5.81)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij}^l y_{ij} \quad \forall (i, j) \in A \quad (5.82)$$

$$x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5.83)$$

$$0 \leq y_{ij} \leq \frac{C_{ij}}{C_{ij}^l} \quad \forall (i, j) \in A \quad (5.84)$$

$CAFWL_l$ と同様に、デザイン変数の上限値である 1 を変更し、 $y_{ij}^s \leq C_{ij}/C_{ij}^l$ とする。 $CAFSL_l$ は線形計画問題であるため、汎用の数値最適化ソフトウェアを用いて解くことができる。しかし、強制制約式 (5.83) を含んでいるため、問題の規模が大きくなると計算時間が大幅に増加する。

$CAFSL_l$ の最適解を \tilde{y} 、 $\tilde{X} (= \sum_{k \in K} \tilde{x}^k)$ とする。パラメータ $\lambda (0 < \lambda < 1)$ 、アーク (i, j) 上の現在の総フロー量 \tilde{X}_{ij} および $l-1$ 回目のアーク (i, j) のアーク容量 C_{ij}^{l-1} を用いて、 l 回目のアーク容量を次のように更新する。

$$C_{ij}^l := \lambda \tilde{X}_{ij} + (1 - \lambda) C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.85)$$

これは、 l 回目のアーク容量を、現在のフロー量と $l-1$ 回目のアーク容量の間に設定することを意味する。 λ によって、 $CAFSL_l$ を解いた結果として得られるフロー量とアーク容量の急激な変化を制御している。

(5.83) 式があるために、最適解において必ずしも $\tilde{y}_{ij} = \tilde{X}_{ij}/C_{ij}^l$ が成り立たず、 $C_{ij}^l \tilde{y}_{ij} \geq \tilde{X}_{ij}$ となり、 $C_{ij}^l \tilde{y}_{ij}$ は \tilde{X}_{ij} の上限値となる。そこで、容量の変更に総フロー \tilde{X} ではなく、 \tilde{X} の上限値である $C_{ij}^l \tilde{y}_{ij}$ を用いる。このとき、アーク (i, j) 上の現在のデザイン変数解 \tilde{y}_{ij} を用いて、 l 回目のアーク容量を次のように更新することもできる。

$$C_{ij}^l := \lambda C_{ij}^{l-1} \tilde{y}_{ij} + (1 - \lambda) C_{ij}^{l-1} = \{1 - \lambda(1 - \tilde{y}_{ij})\} C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.86)$$

これは、 l 回目のアーク容量を、現在のフロー量の上限値である $C_{ij}^{l-1} \tilde{y}_{ij}$ と $l-1$ 回目のアーク容量の間に設定することを意味する。また、この式は容量を $\lambda(1 - \tilde{y}_{ij})$ の比率で増減させることを意味する。

容量スケールリング法では、繰り返し回数とともに多くのデザイン変数は0または1に収束する。しかしながら、すべてのデザイン変数が0または1に収束する保証はないため、繰り返し回数の上限 l_{max} を設定し、一定回数の繰り返しで計算を打ち切る。CAFSL_l に対する基本的な容量スケールリング法の流れを示す。

CAFSL_l に対する基本的な容量スケールリング法

[ステップ1] 繰り返し回数の上限を l_{max} とする。 $C^1 := C$, $l := 0$ とする。

[ステップ2] $l := l + 1$ とする。CAFSL_l を解き、最適解 \tilde{y} , \tilde{x} を求める。

[ステップ3] $C^l := \lambda \tilde{X} + (1 - \lambda)C^{l-1}$ または $C^l := \{1 - \lambda(1 - \tilde{y})\}C^{l-1}$ とする。

[ステップ4] \tilde{y} の成分がすべて0または1であるか、 $l = l_{max}$ であれば終了、そうでなければ、ステップ2へ戻る。

5.3.4 近似解法

CAFW または CAFS に対する容量スケールリング法において、すべてのデザイン変数が0または1となる解が得られれば、デザイン変数が1であるアークで構成されるネットワーク上で、多品種フロー問題を解くことによって、CND の実行可能解と上界値を求めることができる。しかしながら、すべてのデザイン変数が0または1に収束する保証はない。さらに、CAFS では CAFSL_l を解くために多くの計算時間を要することから、十分な繰り返し回数を確保できない場合もある。

そこで、 l 回目の繰り返しにおいて、適当な閾値 $\epsilon_l (0 \leq \epsilon_l \leq 1)$ と、CAFWL_l または CAFSL_l のデザイン解 \tilde{y} を用いて、次のようにデザイン変数 \bar{y} を設定する。

$$\bar{y}_{ij} = \begin{cases} 1 & \text{if } \tilde{y}_{ij} \geq \epsilon_l \\ 0 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad (5.87)$$

続いて、次の多品種フロー問題 $MCF(\bar{y})$ を解く。

(MCF(\bar{y}))

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} \bar{y}_{ij} \quad (5.88)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (5.89)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} \bar{y}_{ij} \quad \forall (i, j) \in A \quad (5.90)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (5.91)$$

MCF(\bar{y}) の最適解を \bar{x} とする。MCF(\bar{y}) が実行可能であれば、 \bar{y} および \bar{x} は CND の実行可能解となり、(5.88) 式に代入すれば CND の上界値が求まる。ただし、必ずしも MCF(\bar{y}) は実行可能解をもつとは限らない。 ϵ_l を大きくすれば、 \bar{y} に0が多くなるため、

実行可能解となる可能性が低くなる．一方， ϵ_l を小さくすれば， \tilde{y} が実行可能解となる可能性は高くなるが，多くのアークを含む解となるため，上界値が大きくなる．

閾値 ϵ_l の変化量を $\alpha(0 < \alpha < 1)$ とする． $MCF(\tilde{y})$ は必ずしも実行可能とは限らないため，実行不可能であれば ϵ_l を α だけ減少させて実行可能性を増加させる．一方，実行可能であれば ϵ_l を α だけ増加させることによって，よりデザイン費用の少ない解を探索可能にする．

近似解法を含めた $CAFW$ および $CAFS$ に対する容量スケールリング法の流れを示す．

CAFW および CAFS に対する容量スケールリング法

[ステップ 1] 繰り返し回数の上限を l_{max} ，閾値の初期値を ϵ_0 ，閾値の変化量を α ，最良の上界値を UB ，上界値を計算する周期を l_{UB} とする． $UB := \infty$ ， $\epsilon_l := \epsilon_0$ ， $C^1 := C$ ， $l := 0$ とする．

[ステップ 2] $l := l + 1$ とする． $CAFWL_l$ または $CAFSL_l$ を解き，最適解 \tilde{y} ， \tilde{x} ， \tilde{X} を求める．

[ステップ 3] $C^l := \lambda \tilde{X} + (1 - \lambda)C^{l-1}$ または $C^l := \{1 - \lambda(1 - \tilde{y})\}C^{l-1}$ とする．

[ステップ 4] $l \bmod l_{UB} = 0$ ， \tilde{y} の成分がすべて 0 または 1，または $l = l_{max}$ であればステップ 5 へ，そうでなければステップ 2 へ戻る．

[ステップ 5]

$$\tilde{y}_{ij} = \begin{cases} 1 & \text{if } \tilde{y}_{ij} \geq \epsilon_l \\ 0 & \text{otherwise} \end{cases} \quad \forall (i, j) \in A$$

として， $MCF(\tilde{y})$ を解き，上界値 UB_l を求める． $UB > UB_l$ であれば， $UB := UB_l$ とする．

[ステップ 6] $MCF(\tilde{y})$ が実行不可能であれば $\epsilon_l := \max(\epsilon_l - \alpha, 0)$ ，そうでなければ $\epsilon_l := \min(\epsilon_l + \alpha, 1)$ とする．

[ステップ 7] \tilde{y} の成分がすべて 0 または 1 であるか， $l = l_{max}$ であれば終了，そうでなければ，ステップ 2 へ戻る．

5.4 パスフローを用いた定式化に対する容量スケールリング法

$CAFS$ に対する容量スケールリング法では，強制制約式 (5.9) を $|K||A|$ 本，フロー変数を $|K||A|$ 個も含む CND の線形緩和問題を繰り返し解く必要があるため，それ自体に多くの計算時間を必要とする．そこで，アークフローではなくパスフローを用いた定式化を用い，必要なパスフローを逐次生成する列生成法と必要な強制制約式を逐次生成する行生成法を組み合わせた解法を示す．

5.4.1 容量スケーリング

$CPFS$ において、デザイン変数の 0-1 条件を 0 から 1 の連続数に緩和した線形緩和問題 $CPFSL$ を考える。 $CPFSL$ において、繰り返し毎にアーク容量を変化させる。繰り返し回数 l のときのアーク (i, j) の容量を C_{ij}^l とする。

パラメータ $\lambda (0 < \lambda < 1)$ 、アーク (i, j) 上の現在の総フロー量 \tilde{X}_{ij} および $l-1$ 回目のアーク (i, j) のアーク容量 C_{ij}^{l-1} を用いて、 l 回目のアーク容量を次のように更新する。

$$C_{ij}^l := \lambda \tilde{X}_{ij} + (1 - \lambda) C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.92)$$

ここで、 $CAFSL_l$ のパスフロー量を \tilde{z} とすると、 \tilde{X}_{ij} は次式となる。

$$\tilde{X}_{ij} = \sum_{k \in K} \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \quad \forall (i, j) \in A \quad (5.93)$$

一方、(5.14) 式があるために、最適解において必ずしも $\tilde{y}_{ij} = \tilde{X}_{ij}/C_{ij}^l$ が成り立たず、 $C_{ij}^l \tilde{y}_{ij} \geq \tilde{X}_{ij}$ となり、 $C_{ij}^l \tilde{y}_{ij}$ は \tilde{X}_{ij} の上限値となる。そこで、アーク (i, j) 上の現在のデザイン変数 \tilde{y}_{ij} を用いて、 l 回目のアーク容量を次のように更新することもできる。

$$C_{ij}^l := \lambda C_{ij}^{l-1} \tilde{y}_{ij} + (1 - \lambda) C_{ij}^{l-1} = \{1 - \lambda(1 - \tilde{y}_{ij})\} C_{ij}^{l-1} \quad \forall (i, j) \in A \quad (5.94)$$

$CPFSL$ の l 回目の繰り返しにおける問題 $CPFSL_l$ は次のようになる。

($CPFSL_l$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.95)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = d^k \quad \forall k \in K \quad (5.96)$$

$$\sum_{k \in K} \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq C_{ij}^l y_{ij} \quad \forall (i, j) \in A \quad (5.97)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5.98)$$

$$z_p^k \geq 0 \quad p \in P^k, k \in K \quad (5.99)$$

$$0 \leq y_{ij} \leq \frac{C_{ij}}{C_{ij}^l} \quad \forall (i, j) \in A \quad (5.100)$$

$CAFWL_l$ と同様に、デザイン変数の上限値である 1 を変更し、 $y_{ij}^s \leq C_{ij}/C_{ij}^l$ に変更している。

5.4.2 限定主問題

取り得るパスの数は非常に多いため、 $CPFSL_l$ は非常に多くのパスフロー変数を含む。そのため、 $CPFSL_l$ を直接解くことは困難である。そこで、あらかじめすべてのパスフ

ロー変数を含む問題を対象とするのではなく、逐次、必要なパスフロー変数を生成し、問題に追加していく。生成するパスフロー変数が単体法の列に相当することから、このような方法を列生成法とよぶ。

一方、 $CPFSL_l$ には、非常に多くの強制制約式 (5.98) が含まれている。しかし、列生成により生成されたパスフロー変数が含まれる強制制約式はそれほど多くなく、生成されたパスフロー変数が左辺に含まれていない強制制約式は不要なものとなる。そこで、生成したパスフロー変数が初めて左辺に含まれる強制制約式を逐次生成し、問題に追加する。生成する制約式が単体法の行に相当することから、このような方法を行生成法とよぶ。

品種 k の適当なパスの部分集合 $\bar{P} = (\bar{P}^k)$ が求められているものとする。このとき、パス集合が \bar{P} に制限されている次のような限定主問題 $CPFSLR_l(\bar{P})$ を考える。

($CPFSLR_l(\bar{P})$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.101)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = d^k \quad \forall k \in K \quad (5.102)$$

$$\sum_{k \in K} \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq C_{ij}^l y_{ij} \quad \forall (i,j) \in A \quad (5.103)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A_{\bar{P}^k}, k \in K \quad (5.104)$$

$$z_p^k \geq 0 \quad p \in \bar{P}^k, k \in K \quad (5.105)$$

$$0 \leq y_{ij} \leq \frac{C_{ij}}{C_{ij}^l} \quad \forall (i,j) \in A \quad (5.106)$$

$A_{\bar{P}^k}$ は品種 k のパス集合 \bar{P}^k に含まれるアーク集合であり、(5.104) 式はアーク (i,j) を通る品種 k のパスフロー変数が生成されているときのみ存在する強制制約式となる。

この問題は線形計画問題であるため、パスの部分集合の要素数が少なければ、汎用の数理最適化ソフトウェアを用いて解くことができる。

5.4.3 列行生成法と行生成法

$CPFSLR_l(\bar{P})$ は変数が限定された問題である。そのため、 $CPFSL_l$ の最適解を求めるためには、逐次、基底に入るであろうパスフロー変数を生成しなければならない。そのために、価格付け問題を解き、被約費用が負であるパスフロー変数を求める。変数に対応するパスを \bar{P}^k に加えて、パスフロー変数を問題に加え、再度 $CPFSLR_l(\bar{P})$ を解き直す。この操作を被約費用が負である変数がなくなるまで繰り返す。被約費用が負である変数がなければ、 $CPFSL_l$ の最適解が得られたことになる。

(5.96) 式に対する双対変数を π 、(5.97)、(5.98) 式に対する非負の双対変数を u 、 w とする。これらの値は、 $CPFSLR_l(\bar{P})$ を最適に解くことにより求めることができる。

パスフロー変数 z に関する被約費用は,

$$\sum_{(i,j) \in A} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k) - \pi^k \quad \forall p \in P^k, k \in K \quad (5.107)$$

である. $\sum_{(i,j) \in A} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k)$ は, アーク (i, j) の長さを $c_{ij}^k + u_{ij} + w_{ij}^k$ としたとき, パス p の長さとなる. また, π^k は現在のパス集合 \bar{P}^k における品種 k の最短距離である. 被約費用は「パス p の長さ - 現在の最短距離」であるので, 被約費用が負である変数を見つけることは現在の最短距離よりも短いパスを見つけることになる.

π^k は定数項として扱えるので, 負のパスフロー変数を見つけるには, 品種 k に対して, (5.107) 式の第一項を最小化するパス p を見つければ良い. したがって, $CPFSLR_l(\bar{P})$ における品種 k に関する価格付け問題は, 次のような問題 PRP^k に帰着される.

(PRP^k)

$$\text{最小化} \quad \sum_{p \in P^k} \left\{ \sum_{(i,j) \in A_{\bar{P}^k}} \delta_{ij}^p (c_{ij}^k + u_{ij} + w_{ij}^k) + \sum_{(i,j) \in A \setminus A_{\bar{P}^k}} \delta_{ij}^p (c_{ij}^k + u_{ij}) - \pi^k \right\} z_p^k \quad (5.108)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad (5.109)$$

$$z_p^k \geq 0 \quad \forall p \in P^k \quad (5.110)$$

ここで, 品種 k の需要に関する強制制約式 (5.98) が生成されていない, すなわち $A_{\bar{P}^k}$ に含まれない制約式に対する w_{ij}^k は 0 とし, (5.108) には含めていない.

π^k は定数であるので, PRP^k は次のようなアーク (i, j) の長さを $c_{ij}^k + u_{ij} + w_{ij}^k$ または $c_{ij}^k + u_{ij}$ とした品種 k に対する始点・終点間の最短路問題 SPP^k に帰着される.

(SPP^k)

$$\text{最小化} \quad \sum_{(i,j) \in A_{\bar{P}^k}} (c_{ij}^k + u_{ij} + w_{ij}^k) \nu_{ij}^k + \sum_{(i,j) \in A \setminus A_{\bar{P}^k}} (c_{ij}^k + u_{ij}) \nu_{ij}^k \quad (5.111)$$

$$\text{条件} \quad \sum_{i \in N_n^+} \nu_{in}^k - \sum_{j \in N_n^-} \nu_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N \quad (5.112)$$

$$\nu_{ij}^k \geq 0, \nu_{ji}^k \geq 0 \quad \forall (i, j) \in A \quad (5.113)$$

品種 $k (\in K)$ について SPP^k を解き, 始点・終点間の最短距離 μ^k を求める. $\nu_{ij}^k = 1$ または $\nu_{ji}^k = 1$ であるアークの集合が最短パス p^k となる. $\mu^k - \pi^k < 0$ であれば, 品種 k における被約費用が負であるパスフロー変数が見つかったことになり, p^k が生成すべきパス, パスに対応するパスフロー変数 $z_{p^k}^k$ が生成すべき変数となる. また, すべての品種 k について $\mu^k - \pi^k \geq 0$ であれば, 被約費用が負である変数が存在しないため, $CPFSL_l$ が最適に解けたことになる.

生成した p^k 上のアークに関して, 新たに $A_{\bar{P}^k}$ の要素となったアーク集合を $\Delta_{\bar{P}^k}$ とする. $\Delta_{\bar{P}^k}$ に含まれるアークについて, 品種 k の需要に関する強制制約式 (5.98) を問題に追加する.

列生成法と行生成法

[ステップ 1] 品種 $k(\in K)$ 毎に適当な初期パス集合 \bar{P}^k を求め, $\bar{P} := (\bar{P}^k)$ とする. 初期パス集合 $\bar{P}^k(k \in K)$ に含まれるパス p 上のアーク (i, j) の集合を $A_{\bar{P}^k}$ とする.

[ステップ 2] $CPFSLR_l(\bar{P})$ を解き, 最適双対解 π, u, w を求める.

[ステップ 3] すべての品種 $k(\in K)$ に対して, 以下の操作を行う.

- アーク $(i, j)(\in A)$ の長さを $c_{ij}^k + u_{ij} + w_{ij}^k$ または $c_{ij}^k + u_{ij}$ とした最短路問題 SPP^k を解き, 品種 k の始点・終点間の最短パス p^k , 最短距離 μ^k を求める.
- $\mu^k - \pi^k < 0$ であれば, 最短パス p^k を \bar{P}^k に加え, パスフロー変数 $z_{p^k}^k$ を生成する. $\Delta_{\bar{P}^k}$ を求める.
- $A_{\bar{P}^k}$ に $\Delta_{\bar{P}^k}$ を加える.
- $\Delta_{\bar{P}^k}$ に含まれるアークに対応する強制制約式を生成し, 制約式として追加する.

[ステップ 4] 追加されたパスがあればステップ 2 へ戻る. そうでなければ終了する.

容量スケールリング法では, 容量を変更して繰り返し線形緩和問題を解く. このため, 二回目以降では, 前回までに生成したパス集合を初期集合として利用できる.

5.4.4 近似解法

アーク容量を変更して, $CPFSL_l$ を繰り返し解き直す. $CPFSL_l$ から得られるデザイン解は線形緩和問題であるため 0 または 1 とは限らないが, 容量スケールリング法の繰り返し回数とともに多くの 0-1 変数である区分変数が 0 または 1 に収束する傾向がある. 一方, 汎用の数値最適化ソフトウェアの分枝限定法を用いると, 少数の 0-1 変数をもつ計画問題に対しては比較的短時間で最適解を求めることができる. しかし, $CPFS$ は $|A|$ 個の 0-1 変数をもつため, 直接解くことは困難である. そこで, 容量スケールリング法において 0 または 1 に収束していないデザイン変数の数がある決められた数以下となった繰り返し回数において, 分枝限定法を適用すれば, 比較的短時間で近似解を求めることができる.

$CPFSL_l$ から得られるデザイン解と適当な閾値を用いて, デザイン変数を 0 に固定, 1 に固定, および固定しない自由変数に分類する. 二つの閾値を ϵ_1 と $\epsilon_2(0 < \epsilon_1 < \epsilon_2 < 1)$ とする. $CPFSL_l$ から得られるデザイン変数解を \tilde{y} とし, ϵ_1 と ϵ_2 を用いて, y を次のように制限する.

$$y_{ij} = \begin{cases} 0 & \text{if } \tilde{y}_{ij} \leq \epsilon_1 \\ 1 & \text{if } \tilde{y}_{ij} \geq \epsilon_2 \end{cases} \quad \forall (i, j) \in A \quad (5.114)$$

$\epsilon_1 < y_{ij} < \epsilon_2$ である 0-1 に固定されていないデザイン変数が決められた数以下となれば, 次のようなデザイン変数とパスが限定された問題 $RFY(\tilde{y}, \bar{P})$ を解く.

$(RFY(\tilde{y}, \bar{P}))$

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (5.115)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = d^k \quad \forall k \in K \quad (5.116)$$

$$\sum_{k \in K} \sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq C_{ij}^l y_{ij} \quad \forall (i,j) \in A \quad (5.117)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A_{\bar{P}^k}, k \in K \quad (5.118)$$

$$z_p^k \geq 0 \quad p \in \bar{P}^k, k \in K \quad (5.119)$$

$$y_{ij} = \begin{cases} 0 & \text{if } \tilde{y}_{ij} \leq \epsilon_1 \\ 1 & \text{if } \tilde{y}_{ij} \geq \epsilon_2 \end{cases} \quad \forall (i,j) \in A \quad (5.120)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \quad (5.121)$$

l 回目の繰り返しにおいて, 0 または 1 に収束していないデザイン変数の数を ω_l , 分枝限定法の実施基準を ψ とする. $\omega_l \leq \psi$ であれば, 収束した変数を固定した問題 $RFY(\tilde{y}, \bar{P})$ に対して分枝限定法を行い, 近似解を算出する. しかし, $RFY(\tilde{y}, \bar{P})$ のように 0-1 変数およびパズフロー変数を限定した問題であっても, 分枝限定法によって最適解を求めることが困難な場合がある. そのため, 実際の計算では, 分枝限定法は一定時間で打ち切り, その時点で得られた最良の上界値を UB_{RFY} とする. ただし, $RFY(\tilde{y}, \bar{P})$ は実行可能でない場合があることに注意する.

ψ を一定値とした場合, 連続した繰り返し回数において分枝限定法を繰り返し同一の解を生成することが多くなる. これを防ぐために分枝限定法を適用する毎に, ψ を $\beta (0 < \beta < 1)$ 倍して減少させる. 最後に, $RFY(\tilde{y}, \bar{P})$ の最適解 \hat{y} を使い, アークフローによる定式化を用いた多品種フロー問題 $MCF(\hat{y})$ を解き直し, 上界値を求める. アークフローによる定式化を用いるのは, パズフローの定式化では, すべてのパスが生成されていないためである.

列生成法および行生成法を用いた容量スケーリング法の流れを示す.

容量スケーリング法

[ステップ 1] 繰り返し回数の上限を l_{max} , 閾値を ϵ_1 および ϵ_2 , 分枝限定法基準を ψ , その変化率を β とする. スケーリングパラメータを λ , 最良の上界値を UB とする. $C^1 := C, l := 0, UB := \infty$ とする.

[ステップ 2] $l := l + 1$ とする. $CPFSL_l$ を解き, 最適解 \tilde{y}, \tilde{z} を求める. $\epsilon_1 < \tilde{y}_{ij} < 1 - \epsilon_2$ を満たすアーク数を ω_l とする.

[ステップ 3] $C^l := \lambda \tilde{X} + (1 - \lambda) C^{l-1}$ または $C^l := \{1 - \lambda(1 - \tilde{y})\} C^{l-1}$ とする.

[ステップ 4] $\omega_l > \psi$ であればステップ 2 へ戻る. そうでなければ, $RFY(\tilde{y}, \bar{P})$ に対して分枝限定法を行い, 最適解 \hat{y} を求める. $\psi := \psi \times \beta$ とする.

[ステップ 5] $MCF(\hat{y})$ を解き, フロー解 \hat{z} および上界値 UB_{MCF} を求める.

$UB_{MCF} < UB$ であれば $UB := UB_{MCF}$ とする .
 [ステップ 6] $l \geq l_{max}$, かつ $UB \neq \infty$ であれば終了する . そうでなければステップ 2
 へ戻る .

ステップ 5 において, $MCF(\hat{y})$ は必ずしも実行可能とは限らず, 上界値が求まらない場合がある . そこで, ステップ 6 で $l \geq l_{max}$ であっても上界値が求まるまで計算を繰り返す .

5.5 数値実験

Crainic らのベンチマーク問題 [36, 37] を使用して, 提案した容量スケールリング法を評価するために数値実験を行う . 単体法にもとづくタブー探索法 [18], 閉路にもとづく近傍探索法 [36], パス再結合法 [37], 多レベル共同タブー探索法 [21] および容量スケールリング法によって求められた上界値を比較する . なお, 使用したデータは www.rku.ac.jp/~katayama/networkdata に掲載している .

主な条件およびパラメータは次の通りである .

- 1) 問題 : Crainic らのベンチマーク問題
- 2) 問題数 : C 問題 43 問, R 問題 153 問
- 3) ノード数 : C 問題 20 ~ 100, R 問題 10 ~ 20
- 4) アーク数 : C 問題 100 ~ 700, R 問題 35 ~ 315
- 5) 品種数 : C 問題 100 ~ 400, R 問題 10 ~ 200
- 6) デザイン費用 : C 問題 $F \cdot V$, R 問題 $c_1 \cdot c_2 \cdot c_8$
- 7) アーク容量 : C 問題 $T \cdot L$, R 問題 $f_{01} \cdot f_{05} \cdot f_{10}$
- 8) 容量の変更方法 : $C^l := \{1 - \lambda(1 - \tilde{y})\}C^{l-1}$
- 9) アークフローに関するパラメータ
 - スケールリングパラメータ λ : 0.050 ~ 0.250 とする .
 - 最大繰り返し回数 l_{max} : 250 回とする .
 - 上界値の計算周期 l_{UB} : 1 回とする .
 - 閾値の初期値 ϵ_0 : 0.5 とする .
 - 閾値の変更量 α : 0.01 とする .
- 10) パスフローに関するパラメータ
 - スケールリングパラメータ λ : 0.050 ~ 0.250 とする .
 - 閾値 ϵ_1, ϵ_2 : $\epsilon_1 = \epsilon_2 = 0.01$ とする .
 - 最大繰り返し回数 l_{max} : 1000 回とする .
 - 分枝限定法の適用基準の初期値 ψ : 50 とする .
 - 分枝限定法の適用基準の変更率 β : 0.75 とする .
 - 分枝限定法の計算時間 : 1 回当り上限時間を 30 分とする .
- 11) 使用計算機 : CPU アークフロー Pentium Core2-Duo 3.3GHz, メモリ 4GB, OS

表 5.1 C 問題:属性

問題	ノード	アーク	品種	費用	容量	問題	ノード	アーク	品種	費用	容量
C1	25	100	10	V	L	C22	100	400	10	V	L
C2	25	100	10	F	L	C23	100	400	10	F	L
C3	25	100	10	F	T	C24	100	400	10	F	T
C4	25	100	30	V	T	C25	100	400	30	V	T
C5	25	100	30	F	L	C26	100	400	30	F	L
C6	25	100	30	F	T	C27	100	400	30	F	T
C7	20	230	40	V	L	C28	30	520	100	V	L
C8	20	230	40	V	T	C29	30	520	100	F	L
C9	20	230	40	F	T	C30	30	520	100	V	T
C10	20	300	40	V	L	C31	30	520	100	F	T
C11	20	300	40	F	L	C32	30	700	100	V	L
C12	20	300	40	V	T	C33	30	700	100	F	L
C13	20	300	40	F	T	C34	30	700	100	V	T
C14	20	230	200	V	L	C35	30	700	100	F	T
C15	20	230	200	F	L	C36	30	520	400	V	L
C16	20	230	200	V	T	C37	30	520	400	F	L
C17	20	230	200	F	T	C38	30	520	400	V	T
C18	20	300	200	V	L	C39	30	520	400	F	T
C19	20	300	200	F	L	C40	30	700	400	V	L
C20	20	300	200	V	T	C41	30	700	400	F	L
C21	20	300	200	F	T	C42	30	700	400	V	T
						C43	30	700	400	F	T

Windows XP

- 12) 使用言語：Visual Studio 2005 C++.NET
- 13) 数理最適化ソフトウェア：CPLEX 11.2
- 14) 上界値の誤差：(各解法による上界値 - 下界値または最良値)/下界値 × 100 の平均値

下界値は CPLEX により求め、最良値は CPLEX または各解法で得られた内の最良値である。CPLEX の計算時間の上限を 10 時間とし、上限を超えた場合はその時点の最良解を採用する。

Crainic らのベンチマーク問題は C 問題と R 問題に分かれている。はじめに C 問題の計算結果を示し、続いて R 問題の計算結果を示す。

C 問題の属性を表 5.1 に示す。表の数値は、ノード数、アーク数、品種数、および費用とアーク容量の性質である。C 問題は合計 43 問である。C 問題は費用およびアーク容量の性質はそれぞれ二種類に分類され、F はデザイン費用がフロー費用に比べて相対的に高い、V はデザイン費用が相対的に安い問題であり、T はアーク容量がきつい、L はアーク

表 5.2 C 問題:解法別平均誤差 従来の解法

解法	誤差 (%)
単体	11.63
サイクル	5.47
パス再結合	5.08
多レベル	4.39

表 5.3 C 問題:解法別平均誤差 容量スケーリング

CAFW		CAFS		CPFS	
パラメータ	誤差 (%)	パラメータ	誤差 (%)	パラメータ	誤差 (%)
0.050	9.19	0.050	3.45	0.050	2.52
0.075	9.40	—	—	0.075	2.61
0.100	9.35	0.100	3.26	0.100	2.65
0.125	8.75	—	—	0.125	2.49
0.150	8.73	0.150	3.15	0.150	2.64
0.175	8.75	—	—	0.175	2.86
0.200	8.86	0.200	3.33	0.200	2.77
0.225	8.57	—	—	0.225	2.74
0.250	9.09	0.250	3.30	0.250	2.92
最良	7.35	最良	2.86	最良	2.21

表 5.4 C 問題:性質別平均誤差

費用	容量	単体 (%)	サイクル (%)	再結合 (%)	多レベル (%)	CAFW (%)	CAFS (%)	CPFS (%)
V	T	5.82	3.42	3.16	2.71	3.44	0.76	0.51
V	L	8.64	3.93	3.42	2.91	4.31	0.93	0.59
F	T	13.88	7.59	6.59	6.01	8.45	5.03	4.37
F	L	17.18	6.42	6.68	5.50	12.48	4.13	2.88
平均		11.63	5.47	5.08	4.39	7.35	2.86	2.21

容量がゆるい問題である。

C 問題に対して、従来の解法である単体法にもとづくタブー探索法 (単体法)、サイクルにもとづくタブー探索法 (サイクル法)、パス再結合法 (再結合法)、多レベル共同タブー探索法 (多レベル法) と容量スケーリング法との比較を行う。提案する三種類の容量スケーリング法を定式化で区別し、*CAFW*、*CAFS*、*CPFS* で表わす。

表 5.2 は、C 問題に対して従来の解法により得られた上界値の平均誤差、表 5.3 は容量ス

表 5.5 C 問題:性質別最良値数

費用	容量	問題数	CAFW	CAFS	CPFS
V	T	10	2	6	7
V	L	10	1	7	8
F	T	12	1	8	9
F	L	11	0	7	8
合計		43	4	28	32

表 5.6 パスフローによる定式化 C 問題:規模別平均誤差

$ N , A , K $	単体 (%)	サイ クル (%)	再結 合 (%)	多レ ベル (%)	CAFW (%)	CAFS (%)	CPFS (%)
25,100, 10	3.29	0.00	0.00	0.03	2.00	1.08	0.53
25,100, 30	1.93	0.78	0.90	0.87	1.63	0.76	0.21
20,230, 40	0.18	0.25	0.20	0.74	0.37	0.21	0.09
20,230,200	33.24	8.89	9.56	6.28	11.74	3.09	2.24
20,300, 40	0.25	0.45	0.43	1.05	0.82	0.27	0.08
20,300,200	23.91	8.99	7.66	6.21	10.87	3.31	2.30
100,400, 10	7.95	4.53	3.43	4.07	9.14	10.21	7.93
100,400, 30	13.60	7.47	6.28	6.83	14.29	7.70	7.49
30,520,100	8.86	6.17	7.00	6.05	9.32	3.31	2.37
30,520,400	14.85	8.96	8.01	5.67	8.23	1.52	1.19
30,700,100	6.47	4.48	4.41	5.06	7.10	1.79	1.44
30,700,400	17.19	11.04	9.40	7.48	10.38	2.42	1.98
平均	11.63	5.47	5.08	4.39	7.35	2.86	2.21

ケーリング法により得られた上界値の平均誤差である。なお、従来の解法の数値は各文献によるものである。パラメータはスケリングパラメータであり、容量スケリング法の「最良」はパラメータを変化させて計算した内の最良値を集計したものである。

単体法の平均誤差は 11.63%、サイクル法の誤差は 5.47%、再結合法の誤差は 5.08% である。従来の解法の中で最も平均誤差の小さな解法は多レベル法であり、4.39% である。一方、CAFW の平均誤差は 8.73~9.40%、最良値は 7.35%、CAFS の平均誤差は 3.15~3.45%、最良値は 2.86% であり、CPFS の平均誤差は 2.49~2.92%、最良値は 2.21% である。CAFW は単体法よりも良いが、他の従来の解法よりも悪い。一方、CPFS と CAFS は従来の解法よりも良い解を算出している。CPFS は CAFS よりも平均誤差が小さいが、これは CPFS には分枝限定法が組み込まれているためである。

表 5.7 C 問題:規模別最良値数

$ N , A , K $	問題数	<i>CAFW</i>	<i>CAFS</i>	<i>CPFS</i>
25,100, 10	3	0	0	0
25,100, 30	3	1	1	2
20,230, 40	3	0	1	2
20,230,200	4	0	4	4
20,300, 40	4	0	1	2
20,300,200	4	0	4	4
100,400, 10	3	1	1	1
100,400, 30	3	0	0	1
30,520,100	4	1	4	4
30,520,400	4	1	4	4
30,700,100	4	0	4	4
30,700,400	4	0	4	4
合計	43	3	28	32

表 5.4は C 問題に対する費用とアーク容量の性質別の平均誤差であり，表 5.5は従来の解法で得られた最良の上界値よりも良い上界値を得た問題数である．いずれの解法でも，デザイン費用が安い V 問題の誤差が小さく，デザイン費用が高い F 問題の誤差が大きい．単体法の誤差は 5.82~17.18%，サイクル法の誤差は 3.42~7.59%，再結合法の誤差は 3.16~6.68%，多レベル法の誤差は 2.71~6.01% である．*CAFW* の誤差は 3.44~12.48% であり，V 問題の誤差は 3.44~4.31% とサイクル法と同程度であるが，F 問題の誤差は 8.45~12.48% とサイクル法よりも劣っている．*CAFS* の誤差は 0.76~5.03% であり，V 問題の誤差は 0.76~0.93% と良い解を算出している．F 問題の誤差は 4.13~5.03% であり，多レベル法よりも 0.98~1.37% 良い解を算出している．*CPFS* の誤差は 0.51~4.37% であり，V 問題の誤差は 0.51~0.59% と良い解を算出している．F 問題の誤差は 2.88~4.37% であり，多レベル法よりも 1.64~2.62% 良い解を算出している．なお，43 問の内，*CAFW* では 4 問，*CAFS* では 28 問，*CPFS* では 32 問で最良値が求められている．

表 5.6は C 問題に対する規模別の平均誤差であり，表 5.7は C 問題に対して最良値を得た問題数である．いずれの解法でも，アーク数または品種数が増加すると誤差が大きくなる傾向がある．単体法の誤差は 0.18~33.24%，サイクル法の誤差 0.00~11.04%，再結合法の誤差は 0.00~9.56%，多レベル法の誤差は 0.00~7.48% である．*CAFW* の誤差は 0.37~14.29% であり，サイクル法よりも劣っている．*CAFS* の誤差は 0.21~10.21% であり，ノード数が多い 100 ノードの問題の誤差は 7.70~10.21% であり，多レベル法に大きく劣るが，ノード数が少ない問題では良い解を算出している．*CPFS* の誤差は 0.08~7.93% であり，*CAFS* と同様の傾向があり，ノード数が多い 100 ノードの問題の誤差は

表 5.8 C 問題:個別結果 1

問 題	最適値 または 下界値	従来法 による 上界値	CAFW		CAFS		CPFS		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
C1	14712 ^O	14712	14712	0.00	14712	0.00	14712	0.00	0.00
C2	14941 ^O	14941	15231	1.94	15037	0.64	15012	0.48	0.48
C3	49899 ^O	49899	51922	4.05	51199	2.61	50458	1.12	1.12
C4	365272 ^O	365272	365695	0.12	365282	0.00	365272	0.00	0.00
C5	37055 ^O	37055	38547	4.03	37489	1.17	37284	0.62	-0.62
C6	85530 ^O	85530	86166	0.74	86488	1.12	85530	0.00	-0.89
C7	423848 ^O	423848	425508	0.39	425361	0.36	424075	0.05	-0.07
C8	371475 ^O	371475	371944	0.13	372232	0.20	371906	0.12	0.12
C9	643036 ^O	643036	646907	0.60	643555	0.08	643666	0.10	-0.08
C10	429398 ^O	429398	429912	0.12	429398	0.00	429398	0.00	0.00
C11	586077 ^O	586077	595699	1.64	592176	1.04	587800	0.29	-0.24
C12	464509 ^O	464509	465331	0.18	464659	0.03	464569	0.01	0.01
C13	604198 ^O	604198	612228	1.33	604208	0.00	604198	0.00	-0.36
C14	92598 ^L	94386	100591	8.63	95608	3.25	94247	1.78	-4.40
C15	133512 ^L	141737	158213	18.50	139010	4.12	138123	3.45	-3.51
C16	97344 ^L	97914	104896	7.76	98873	1.57	97968	0.64	-3.98
C17	132432 ^L	137271	148400	12.06	136973	3.43	136497	3.07	-3.32
C18	73759 ^L	74972	81011	9.83	75539	2.41	74995	1.68	-4.08
C19	111655 ^L	117306	128089	14.72	117754	5.46	115876	3.78	-4.98
C20	74991 ^O	74991	80705	7.62	75603	0.82	75302	0.41	-2.52
C21	104334 ^L	108252	116120	11.30	109083	4.55	107793	3.31	-3.04

O:最適値; L:下界値

7.49~7.93%であり、多レベル法に劣っている。

表 5.8および表 5.9は、C 問題の個別の結果の比較である。数値は、CPLEX による最適値または下界値、従来法により得られた最良の上界値、CPFS による上界値である。なお、従来法は、単体法、サイクル法、パス再結合法、多レベル法である。「最適値または下界値」の列における O は最適値、L は下界値であることを表す。誤差は最適値または下界値との差であり、改善率は従来法最良値との差である。改善率がマイナスである問題は、容量スケール法により従来法最良値よりも良い解が見つかったことを表す。CAFW の誤差は 0.00~26.70% であり、アーク数の大きな FL 問題で誤差が大きくなり、C26 の誤差は 26.70% である。CAFS の誤差は 0.00~24.05% であり、アーク数が多い C24、C26、C27 の誤差が大きい。CPFS の誤差は 0.0~21.65% であり、CAFS

表 5.9 C 問題:個別結果 2

問 題	最適値 または 下界値	従来法 による 上界値	CAFW		CAFS		CPFS		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
C22	28423 ^O	28423	28423	0.00	28423	0.00	28426	0.01	-0.21
C23	23949 ^O	24436	26507	10.68	25526	6.58	24459	2.13	2.13
C24	59470 ^L	66364	69423	16.74	73772	24.05	72345	21.65	10.83
C25	384560 ^L	385544	387736	0.83	384940	0.10	384883	0.08	-0.01
C26	47459 ^L	50496	59895	26.20	52551	10.73	51956	9.48	2.97
C27	127825 ^L	141278	148087	15.85	143523	12.28	144314	12.90	2.09
C28	53958 ^O	53966	57068	5.76	54211	0.47	54065	0.20	-1.53
C29	91285 ^L	95294	107947	18.25	96779	6.02	94872	3.93	-4.73
C30	51825 ^L	52085	52984	2.24	52455	1.22	52282	0.88	-1.33
C31	94646 ^L	98357	105076	11.02	99883	5.53	98888	4.48	-3.50
C32	47603 ^O	47603	50225	5.51	47671	0.14	47680	0.16	-1.48
C33	58772 ^L	60525	65610	11.64	60496	2.93	60116	2.29	-3.77
C34	45552 ^L	45945	47520	4.32	46293	1.63	46169	1.35	-1.82
C35	54233 ^L	55709	57990	6.93	55561	2.45	55301	1.97	-2.25
C36	111992 ^L	112998	118115	5.47	113358	1.22	112846	0.76	-2.44
C37	146809 ^L	150270	163982	11.70	150261	2.35	149465	1.81	-4.56
C38	114237 ^L	114793	119723	4.80	114743	0.44	114641	0.35	-4.60
C39	150009 ^L	163352	166450	10.96	153127	2.08	152783	1.85	-4.64
C40	96741 ^L	98234	103892	7.39	98176	1.48	97974	1.27	-4.54
C41	130724 ^L	135337	154162	17.93	136489	4.41	135205	3.43	-6.10
C42	94118 ^L	95524	100143	6.40	95593	1.57	95306	1.26	-3.92
C43	127666 ^L	133402	140199	9.82	130505	2.22	130148	1.94	-5.87

O:最適値; L:下界値

と同様の傾向がある。CPFS により 43 問の内の 32 問で従来の最良値を改善しており、最大で 6.10% 改善している。

表 5.10 および表 5.11 は、従来の解法と容量スケール法による C 問題の性質別と規模別の平均計算時間である。容量スケール法では PC (CPU 3.3GHz, RAM 4GBytes) を使用している。一方、単体法では SUN Ultra60/2300 (2CPUs 296MHz (one CPU use), RAM 2GBytes), サイクル法 および再結合法では SUN Enterprise 10000 (64CPUs 400MHz (one CPU use), RAM 64GBytes) を使用している。このように、使用しているコンピュータが異なるため、単純に計算時間を比較することはできない。単体法の全体の平均計算は 2273.8 秒、サイクル法の平均計算は 9327.0 秒、再結合法の平均計算は

表 5.10 C 問題:性質別平均計算時間

費用	容量	単体 (秒)	サイクル (秒)	再結合 (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
V	T	2806.3	8082.8	5792.0	24.8	2544.7	32.9
V	L	2235.4	7721.7	5938.3	45.8	1764.8	46.1
F	T	2089.0	9351.3	9065.0	68.7	4326.1	88.7
F	L	2026.3	11891.1	11203.5	53.2	7982.5	138.2
平均		2273.8	9327.0	8123.7	49.2	4251.5	78.5

表 5.11 C 問題:規模別平均計算時間

$ N , A , K $	単体 (秒)	サイクル (秒)	再結合 (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
025,100,010	10.4	51.3	16.9	0.3	0.5	0.5
025,100,030	40.5	221.2	91.2	0.6	1.9	0.7
020,230,040	94.5	409.8	159.3	1.2	1.7	0.2
020,230,200	608.5	2766.9	2742.5	24.3	1676.4	176.4
020,300,040	113.3	585.9	228.9	1.5	2.0	0.3
020,300,200	1075.8	4230.0	3840.0	39.3	2697.4	146.6
100,400,010	49.0	423.2	127.3	1.3	26.2	12.4
100,400,030	197.8	1715.3	429.5	10.6	351.5	18.6
030,520,100	955.9	3699.2	1422.6	12.2	884.9	53.7
030,520,400	6783.3	36921.3	34852.3	276.6	10419.4	179.4
030,700,100	1726.6	4644.4	1824.6	10.9	401.7	8.7
030,700,400	12886.0	45302.4	41801.3	153.7	29335.9	254.2
平均	2273.8	9327.0	8123.7	49.2	4251.5	78.5

8123.7秒である。また、CAFWの全体の平均計算時間は49.2秒、CAFSは4251.5秒、CPFSは78.5秒である。CAFWとCPFSの計算時間は短く、CPFSの平均計算時間はCAFSの1/50以下となっており、列生成法と行生成法により大幅に計算時間を短縮することができる。

性質別では、CAFWの計算時間は24.8~68.7秒、CAFSの計算時間は24.8~7982.5秒、CPFSの計算時間は32.9~138.2秒であり、性質別の計算時間では顕著な特徴は見られない。問題の規模別では、CAFWの計算時間は0.3~276.6秒、CAFSの計算時間は0.5~29335.9秒、CPFSの計算時間は0.2~254.2秒である。アーク数が大きく、品種数が多くなると計算時間が増大し、特に400品種の問題では計算時間が大きい。

表 5.12および表 5.13は、個別のC問題の計算時間の比較である。 t はCPLEXの分枝

表 5.12 C 問題:個別計算時間 1

問題	最適/下界 (秒)	単体 (秒)	サイクル (秒)	再結合 (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
C1	0.2	5.6	48.9	12.5	0.2	0.3	0.1
C2	11.1	8.4	53.8	14.1	0.3	0.8	0.9
C3	2.9	17.1	51.2	24.1	0.3	0.5	0.5
C4	0.6	16.6	223.7	101.4	0.5	0.7	0.2
C5	182.5	33.0	215.4	75.2	0.7	3.5	1.2
C6	46.5	71.8	224.6	97.0	0.6	1.4	0.5
C7	1.7	71.3	370.3	148.8	1.1	1.5	0.2
C8	9.0	90.3	435.6	156.9	1.3	1.7	0.2
C9	30.1	121.8	423.3	172.2	1.3	1.8	0.2
C10	1.1	71.1	611.5	224.9	1.4	0.7	0.2
C11	23.0	113.4	581.9	228.3	1.6	2.7	0.4
C12	12.7	145.3	589.6	247.9	1.5	2.2	0.3
C13	11.8	123.4	560.4	214.4	1.6	2.2	0.4
C14	t	504.5	2663.2	2494.9	13.1	1005.3	81.3
C15	t	491.6	2718.3	2878.3	22.3	2371.4	383.9
C16	t	548.4	2565.7	2210.9	31.0	1253.9	82.7
C17	t	889.7	3120.1	3385.8	30.6	2074.9	157.8
C18	t	982.2	4086.8	3566.0	44.6	1445.4	92.8
C19	t	1316.8	4367.9	4012.6	61.8	4527.1	259.6
C20	35575.5	938.3	3807.9	3924.2	26.6	1213.8	100.2
C21	t	1065.9	4657.5	3857.1	24.0	3603.4	133.9

 t :計算時間の上限

限定法が計算時間の上限である 10 時間の制限で終了し、 $t(-)$ は CPLEX により実行可能解を求められなかったことを表す。CPLEX により、小さな問題では最適解を求めることができたが、400 品種の大きな問題では実行可能解を求めることができない場合が多い。CAFW の計算時間は 0.2 ~ 627.3 秒であり、最大 10 分程度である。CAFS の計算時間は 0.3 ~ 53431.2 秒、最大 15 時間程度であり、特に 400 品種の規模の大きな問題で計算時間が大きい。CPFS の計算時間は 0.1 ~ 436.3 秒、最大 7 分間程度であり、CAFW と同程度の計算時間である。CPU などが異なるため、計算時間を直接比較することはできないが、CAFW と CPFS は従来の解法に比べて、極めて短時間で解を算出している。

R 問題の 153 問に対して、容量スケールリング法と従来の解法である単体法、サイクル法、パス再結合法との比較を行う。R 問題の属性を表 5.14 および表 5.15 に示す。表の数値は、ノード数、アーク数、品種数、アーク容量と費用の性質である。R 問題では、アー

表 5.13 アークフローによる定式化 C 問題:計算時間 2

問題	最適/下界 (秒)	単体 (秒)	サイクル (秒)	再結合 (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
C22	2.3	32.7	336.3	89.2	89.2	0.6	0.2
C23	4752.0	33.0	306.8	82.9	1.1	49.8	34.3
C24	t	81.2	626.5	209.9	2.5	27.9	2.6
C25	t	277.5	1975.3	492.8	0.3	6.0	0.6
C26	t	100.2	1300.6	315.0	2.6	619.6	31.0
C27	t	215.7	1870.0	480.9	29.0	428.9	24.3
C28	12725.2	995.6	3356.0	1194.1	6.9	45.5	2.9
C29	t	939.2	4032.4	1460.0	21.4	1810.0	140.6
C30	t	1218.5	3481.1	1513.7	7.3	71.5	1.6
C31	t	670.3	3927.4	1522.7	13.2	1612.4	69.8
C32	402.4	1265.1	4396.4	1860.6	9.0	30.3	3.0
C33	t	1479.6	4755.0	1837.5	10.0	521.0	15.7
C34	t	2426.0	4560.1	1894.1	10.1	259.1	4.1
C35	t	1735.7	4866.1	1706.1	14.5	796.3	12.1
C36	t	5789.3	36530.8	27477.4	352.9	4925.2	146.5
C37	$t(-)$	6406.6	42929.6	36669.3	40.6	24470.6	216.7
C38	$t(-)$	6522.2	28214.0	23089.1	85.7	3748.0	26.2
C39	$t(-)$	8415.2	40010.9	52173.2	627.3	8533.6	328.2
C40	$t(-)$	12636.2	24816.8	22314.8	28.8	10192.5	133.6
C41	$t(-)$	11367.7	69540.1	75664.9	422.7	53431.2	436.3
C42	$t(-)$	15879.5	34974.9	24288.9	84.0	18890.0	113.1
C43	$t(-)$	11660.4	51877.9	44936.4	79.4	34830.0	334.0

t :計算時間の上限; $-$:CPLEX で実行可能解なし

ク容量は三種類に分類され, $c1$ はアーク容量が大きい, $c8$ はアーク容量が小さいことを表す. また, 費用は三種類に分類され, $f01$ はデザイン費用が相対的に安い, $f10$ はデザイン費用が相対的に高いことを表す.

表 5.16および表 5.17は R 問題をアーク容量と費用の組合せによって分類し, 従来の解法と $CPFS$ を比較したものである. 表 5.16は CPLEX による最適値または上界値と近似解法による上界値の誤差であり, 表 5.17は CPLEX による最適値または下界値と近似解法による上界値の誤差である. 本来の誤差は後者であるが, 従来の研究では前者を用いているため, 併記している. 単体法の下界値との平均誤差は 9.95%, サイクル法の平均誤差は 4.01% である. 再結合法の平均誤差は個別の結果が文献に記載されていないため, 記載していない. 一方, $CAFW$ の平均誤差は 5.48%, $CAFS$ の平均誤差は 1.93%,

CPFS の平均誤差は 1.09% である。

性質別にみると、単体法の下界値との誤差は 2.28~22.28%，サイクル法の誤差は 1.38~6.28% である。一方、*CAFW* の誤差は 2.10~9.89%，*CAFS* の誤差は 0.10~3.82%，*CPFS* の誤差は 0.05~2.35% である。いずれの解法でもデザイン費用の高い問題の誤差が大きい。また、すべての分類で *CPFS* が最も良い解を算出している。

表 5.18 は、R 問題に対して最良値を得た問題数である。*CAFW* は 153 問題内の 47 問題、*CAFS* は 101 問題で最良値を算出している。また、*CPFS* は、各分類の内の 60% 以上である 111 問題で最良値を算出している。なお、*CAFW* は 153 問題内の 21 問題、*CAFS* は 38 問題、*CPFS* は 49 問題で最適解を算出している。

表 5.19 および表 5.20 は、R 問題をノード数、アーク数、品種数によって分類し、従来の解法と容量スケールリング法を比較したものである。表 5.19 は CPLEX による最適値または上界値と近似解法による上界値の誤差であり、表 5.20 は CPLEX による最適値または下界値と近似解法による上界値の誤差である。単体法の下界値との誤差は 0.57~30.32%，サイクル法の誤差は 0.00~14.07%，*CAFW* の誤差は 0.14~13.67%，*CAFS* の誤差は 0.11~7.86%，*CPFS* の誤差は 0.11~3.51% である。いずれの解法でもアーク数が多い、品種数が多い問題の誤差が大きい。また、すべての分類で *CPFS* が最も良い解を算出している。

表 5.21 は、R 問題に対して最良値を得た問題数である。*CPFS* は、10~25 品種の小さな問題に対して新たな最良値を算出できていない場合が多い。しかし、アーク数が 60 以上かつ品種数が 40 以上の問題に対しては、9 問題のうち 8~9 問題で最良値を算出している。

表 5.22 および表 5.23 は、従来の解法と容量スケールリング法による R 問題の性質別と規模別の平均計算時間である。単体法の平均計算時間は 1575.2 秒、サイクル法の平均計算時間は 789.8 秒である。また、*CAFW* の平均計算時間は 9.9 秒、*CAFS* は 288.0 秒、*CPFS* は 47.3 秒である。*CAFS* の平均計算時間は、*CAFW* の 5 倍程度、*CAFS* の 1/6 程度である。性質別では、*CAFW* の計算時間は 2.4~21.2 秒、*CAFS* の計算時間は 18.9~577.0 秒、*CPFS* の計算時間は 1.7~89.8 秒である。規模別では、*CAFW* の計算時間は 0.1~72.8 秒、*CAFS* の計算時間は 0.1~2399.2 秒、*CPFS* の計算時間は 0.0~401.0 秒である。容量スケールリング法では、10 ノードの問題では 2 秒以内で解を算出している。一方、品種が多い、アーク数が多い問題の計算時間が大きい。

個別の R 問題の計算結果の比較を付録の表 A.1~A.6 に、個別の R 問題の計算時間の比較を表 A.7~A.9 に示す。

表 5.14 R 問題:属性 1

問題	ノード	アーク	品種	問題	ノード	アーク	品種
R01	10	35	10	R10	20	120	40
R02	10	35	25	R11	20	120	100
R03	10	35	50	R12	20	120	200
R04	10	60	10	R13	20	220	40
R05	10	60	25	R14	20	220	100
R06	10	60	50	R15	20	220	200
R07	10	83	10	R16	20	315	40
R08	10	83	25	R17	20	315	100
R09	10	83	50	R18	20	315	200

表 5.15 R 問題:属性 2

問題	容量	費用
R*-1	c1	f01
R*-2	c1	f05
R*-3	c1	f10
R*-4	c2	f01
R*-5	c2	f05
R*-6	c2	f10
R*-7	c8	f01
R*-8	c8	f05
R*-9	c8	f10

5.6 本章のまとめ

本章では、容量制約のあるネットワーク設計問題に対して、三種類の容量スケールリング法を用いた近似解法を提案した。アークフロー変数を用いた定式化では、強制制約式を含まない弱い定式化と強制制約式を含む強い定式化について容量スケールリング法を適用した。パスフロー変数を用いた定式化では、強制制約式を含む強い定式化について容量スケールリング法を適用した。また、Crainicらのベンチマーク問題を用いて数値実験を行い、提案した解法の有効性を示した。

本章では、容量制約をもつネットワークデザイン問題に対して、以下のような成果を示した。

- 1) アークフローを用いた定式化を示した。

表 5.16 R 問題:性質別平均誤差 1*¹

容量	費用	単体 (%)	サイクル (%)	再結合 (%)	CAFW (%)	CAFS (%)	CPFS (%)
c1	f01	2.49	1.48	0.76	2.71	0.09	0.04
c1	f05	12.31	3.49	2.43	7.87	1.07	0.22
c1	f10	19.86	3.55	3.09	8.09	1.19	0.01
c2	f01	2.21	1.31	0.78	2.03	0.22	0.09
c2	f05	9.16	3.68	2.64	6.00	1.02	0.35
c2	f10	14.45	4.27	3.04	6.08	2.16	0.18
c8	f01	2.96	1.74	1.15	1.39	0.51	0.33
c8	f05	6.33	4.14	3.23	2.21	1.68	0.60
c8	f10	8.60	4.40	4.11	3.77	1.82	0.64
平均		8.87	3.10	—	4.58	1.07	0.26

*1:CPLEX による最適値/上界値との誤差

表 5.17 R 問題:性質別平均誤差 2*²

容量	費用 (%)	単体 (%)	サイクル (%)	CAFW (%)	CAFS (%)	CPFS (%)
c1	f01	2.49	1.48	2.72	0.10	0.05
c1	f05	13.00	4.03	8.44	1.57	0.72
c1	f10	22.28	5.28	9.89	2.83	1.61
c2	f01	2.28	1.38	2.10	0.28	0.16
c2	f05	10.00	4.42	6.74	1.70	1.02
c2	f10	16.64	5.96	7.75	3.82	1.72
c8	f01	3.30	2.07	1.71	0.82	0.64
c8	f05	7.66	5.43	3.44	2.90	1.78
c8	f10	10.54	6.28	5.60	3.61	2.35
平均		9.95	4.01	5.48	1.93	1.09

*2:CPLEX による最適値/下界値との誤差

強制制約式を含まない弱い定式化および強制制約式を含む強い定式化を示した。
また、これらの定式化に対する二種類の線形緩和問題と二種類の Lagrange 緩和問題を
示し、問題の性質と解法を示した。

2) パスフローを用いた定式化を示した。

強制制約式を含むパスフローを用いた強い定式化を示した。この定式化の線形緩和
問題を示し、線形緩和問題におけるパスフロー変数の被約費用と双対問題を示

表 5.18 R 問題:性質別最良値数

容量	費用	問題数	CAFW	CAFS	CPFS
c1	f01	18	2	12	11
c1	f05	18	3	12	14
c1	f10	18	2	10	13
c2	f01	18	2	11	11
c2	f05	18	5	11	13
c2	f10	18	5	11	13
c8	f01	15	9	11	10
c8	f05	15	12	11	13
c8	f10	15	7	12	13
合計		153	47	101	111

した。

- 3) アークフローを用いた定式化に対して容量スケールリング法を提案した。

弱い定式化に対する容量スケールリング法および強い定式化に対する容量スケールリング法を示した。また、容量スケールリング法によって得られた解と閾値を用いて近似解を求める近似解法を示した。

- 4) パスフローを用いた定式化に対して容量スケールリング法を示した。

パスフローを用いた定式化に対する容量スケールリング法を示した。線形緩和問題に対して、パスフロー変数を限定した限定主問題を示した。また、限定主問題における価格付け問題を示し、この問題が最短路問題となることを示した。逐次、パスフロー変数と強制制約式を付加する列生成法と行生成法を示した。さらに、容量スケールリング法によって得られた解をもとにした分枝限定法を示した。

- 5) 数値実験により、提案する解法の有効性を示した。

Crainicらのベンチマーク問題を用いて、従来の解法と容量スケールリング法を比較した。アークフローを用いた強い定式化に対する容量スケールリング法およびパスフローを用いた定式化に対する容量スケールリング法では、従来の解法に比べて多くの問題で良い解を求めることができた。パスフローを用いた定式化に対する容量スケールリング法では、列生成法により大幅に計算時間を短縮することができた。

- 6) 問題の属性と適用すべき解法

ノード数が大きな問題に対しては多レベル共同タブー探索法を適用し、それ以外の問題に対しては提案したCPFSを適用することが良いと考える。また、短時間で解を求めたい場合は、問題の属性によらず提案したCPFSを適用することが良いと考える。

今後の課題としては、次の項目が挙げられる。

表 5.19 R 問題:規模別平均誤差 1*¹

$ N , A , K $	単体 (%)	サイクル (%)	再結合 (%)	CAFW (%)	CAFS (%)	CPFS (%)
10,35 ,10	0.62	0.00	0.00	0.64	0.59	0.59
10,35 ,25	0.57	0.78	0.23	0.93	0.13	0.11
10,35 ,50	0.54	1.63	0.61	0.14	0.11	0.11
10,60 ,10	1.81	0.11	0.08	1.23	0.95	0.52
10,60 ,25	2.78	0.48	0.36	1.56	0.80	0.34
10,60 ,50	7.69	2.47	1.14	4.62	0.32	0.21
10,83 ,10	2.04	0.41	0.04	1.40	1.13	0.83
10,83 ,25	4.03	0.91	0.41	4.29	0.93	0.76
10,83 ,50	8.96	2.87	1.52	4.64	0.82	0.36
20,120,40	5.00	2.78	1.37	5.73	1.15	0.63
20,120,100	9.03	2.69	2.05	3.22	1.10	0.16
20,120,200	8.06	5.12	4.55	2.23	0.76	0.17
20,220,40	8.43	3.51	3.59	8.53	2.71	1.54
20,220,100	16.87	6.33	4.93	7.88	0.87	0.01
20,220,200	24.41	6.82	5.41	4.98	0.36	-0.48
20,315,40	6.47	2.90	2.08	9.50	1.94	0.68
20,315,100	20.81	5.47	4.68	10.53	1.53	-0.11
20,315,200	23.24	8.20	6.84	6.37	2.23	-1.77
平均	8.87	3.10	—	4.58	1.07	0.26

*1:CPLEX による最適値/上界値と近似解法による上界値の誤差

1) 新たな妥当不等式の提示

品種の需要に対する強制制約式は有効な妥当不等式となるが、依然、問題によっては大きな双対ギャップが存在する。そのため、新たな有効な妥当不等式の開発が必要である。

2) ノード数が大きな問題に対する精度

パスフローを用いた定式化に対する容量スケール法により、従来の解法から得られる解に比べて多くの問題で良い解を求めることができた。しかし、ノード数が大きな問題に関しては、従来の解法が優れている場合があり、これらの要因の解析と解法へのフィードバックが必要である。

3) 分枝限定法への列生成法の組み込み

パスフローを用いた定式化の分枝限定法では、既に生成された変数のみを用いている。このため、分枝限定法に移行した限定されたデザイン変数に対する最適解を求めるためには、分枝限定法内で列生成を行うことが必要である。

表 5.20 R 問題:規模別平均誤差 2*2

$ N , A , K $	単体 (%)	サイクル (%)	CAFW (%)	CAFS (%)	CPFS (%)
10,35,10	0.62	0.00	0.64	0.59	0.59
10,35,25	0.57	0.78	0.93	0.13	0.11
10,35,50	0.54	1.63	0.14	0.11	0.11
10,60,10	1.81	0.11	1.23	0.95	0.52
10,60,25	2.78	0.48	1.56	0.80	0.34
10,60,50	7.69	2.47	4.62	0.32	0.21
10,83,10	2.04	0.41	1.40	1.13	0.83
10,83,25	4.03	0.91	4.29	0.93	0.76
10,83,50	8.96	2.87	4.64	0.82	0.36
20,120,40	5.00	2.78	5.73	1.15	0.63
20,120,100	9.07	2.73	3.26	1.14	0.19
20,120,200	8.58	5.61	2.69	1.20	0.60
20,220,40	8.78	3.86	8.87	3.04	1.88
20,220,100	18.84	8.07	9.63	2.51	1.63
20,220,200	28.03	9.51	7.63	2.84	1.96
20,315,40	7.73	4.14	10.78	3.16	1.87
20,315,100	24.40	8.46	13.67	4.41	2.69
20,315,200	30.32	14.07	12.09	7.86	3.51
平均	9.95	4.01	5.48	1.93	1.09

*2:CPLEX による最適値/下界値との誤差

表 5.21 R 問題:規模別最良値数

$ N , A , K $	問題数	<i>CAFW</i>	<i>CAFS</i>	<i>CPFS</i>
10,35,10	6	0	0	0
10,35,25	6	2	3	3
10,35,50	6	2	2	2
10,60,10	9	1	1	1
10,60,25	9	1	2	4
10,60,50	9	1	8	8
10,83,10	9	0	1	1
10,83,25	9	1	3	3
10,83,50	9	3	9	9
20,120,40	9	3	8	9
20,120,100	9	6	8	9
20,120,200	9	9	9	9
20,220,40	9	1	6	9
20,220,100	9	4	9	9
20,220,200	9	6	9	9
20,315,40	9	0	6	8
20,315,100	9	2	9	9
20,315,200	9	5	8	9
合計	153	47	101	111

表 5.22 R 問題:性質別平均計算時間

容量	費用	単体 (秒)	サイクル (秒)	<i>CAFW</i> (秒)	<i>CAFS</i> (秒)	<i>CPFS</i> (秒)
c1	f01	563.5	373.5	2.4	18.9	1.7
c1	f05	683.0	363.0	4.1	412.4	69.8
c1	f10	694.7	323.7	6.0	703.7	88.3
c2	f01	745.2	619.0	4.3	87.4	8.5
c2	f05	1259.9	476.8	10.6	577.0	89.8
c2	f10	1039.5	422.8	14.9	411.6	74.2
c8	f01	2675.7	1657.3	21.2	55.6	30.0
c8	f05	3639.1	1791.0	13.6	81.2	23.0
c8	f10	3769.6	1580.3	15.8	147.6	30.4
平均		1575.2	789.8	9.9	288.0	47.3

表 5.23 R 問題:規模別平均計算時間

$ N , A , K $	単体 (秒)	サイクル (秒)	<i>CAFW</i> (秒)	<i>CAFS</i> (秒)	<i>CPFS</i> (秒)
10,35 ,10	9.9	6.8	0.1	0.1	0.0
10,35 ,25	27.3	22.0	0.2	0.2	0.1
10,35 ,50	59.1	42.4	0.3	0.3	0.1
10,60 ,10	15.0	10.9	0.1	0.1	0.1
10,60 ,25	43.7	31.4	0.3	0.4	0.2
10,60 ,50	87.0	63.8	0.4	1.4	0.6
10,83 ,10	18.0	13.3	0.1	0.2	0.1
10,83 ,25	49.3	33.3	0.3	0.6	0.4
10,83 ,50	97.6	69.1	0.7	1.8	0.8
20,120,40	231.7	212.1	0.9	3.8	2.3
20,120,100	851.8	780.7	4.4	36.6	9.6
20,120,200	4139.0	2052.9	12.2	81.0	30.4
20,220,40	337.9	260.0	1.9	31.1	7.1
20,220,100	1574.4	861.5	11.4	184.8	56.5
20,220,200	6938.9	3599.1	49.0	1405.3	174.6
20,315,40	419.5	290.2	2.3	42.8	33.0
20,315,100	1412.3	1187.1	11.4	706.3	86.6
20,315,200	10498.6	3834.1	72.8	2399.2	401.0
平均	1575.2	789.8	9.9	288.0	47.3

第 6 章

区分的線形費用をもつネットワーク設計問題の近似解法

6.1 本章の概要

輸送やロジスティクスなどの分野における現実的な問題では、宅配便、混載便、貸切便といったような複数種類の輸送手段を選択することができ、処理量によって費用が最小となる輸送手段を利用することになる。そのため、費用は処理量に対して区分的線形である非線形関数として表現できる。このような問題は、区分的線形費用をもつネットワーク設計問題 (PND) とよばれる。

PND は、より現実的な費用を考慮した問題であるため、容量制約のあるネットワーク設計問題である CND と比べてもさらに複雑な問題であり、それほど規模の大きくない問題であっても適切な解を求めることが困難な場合がある。近年研究が始まったばかりであることもあって、いまだ一般的に普及している解法が提供されていない状況にある。また、従来の解法では実行可能解を算出できない場合があり、ある程度の規模までの問題に対して安定して実行可能解を算出することができる解法が必要となる。

本章では、 PND に対して、アークフローおよびパスフローを用いた定式化を示す。続いて、アークフローおよびパスフローを用いた定式化に対する容量スケールリング法を提案し、数値実験によって提案した解法の有効性を検討する。最後に、本章のまとめを記述する。

6.2 問題の定式化

はじめに、 PND の前提条件、使用する記号と PND の定義を示す。続いて、 PND に対する三種類の定式化、およびパスフローを用いた拡張した定式化を示す。

6.2.1 前提条件，記号および問題の定義

前提条件

- ノード集合が与えられている．
- アーク集合が与えられている．
- アークは向きをもつ．
- アークには，フロー量に対する区分的線形関数であるフロー費用関数が与えられている．
- 区分的線形関数は，下半連続である．
- 区分的線形関数における各線形関数は，固定費用と変動費用で表現される．固定費用および単位当たりの変動費用は非負である．
- 複数の品種からなる品種集合が与えられている．
- 各品種の需要が与えられている．
- 各品種の需要は，始点から終点までのパス上を移動する．

記号の定義

- N : ノード集合
- A : アーク集合
- K : 品種集合
- N_n^+ : ノード n を始点とするアークの終点であるノード集合
- N_n^- : ノード n を終点とするアークの始点であるノード集合
- P^k : 品種 k の取り得るパス集合
- \bar{P}^k : 品種 k の取り得るパス集合の部分集合
- \bar{P} : \bar{P}^k の k に関する和集合
- $A_{\bar{P}^k}$: \bar{P}^k に含まれる品種 k のパスに含まれるアーク集合
- S_{ij} : アーク (i, j) における区分集合
- c_{ij}^s : アーク (i, j) の区分 s における単位フロー量当たりの非負の変動費用
- f_{ij}^s : アーク (i, j) の区分 s における非負の固定費用
- b_{ij}^s : アーク (i, j) の区分 s の範囲の上限値，区分 $s + 1$ の範囲の下限値
- b_{ij}^{sl} : l 回目の繰り返しにおけるアーク (i, j) の区分 s に対応する容量
- d^k : 品種 k の需要
- δ_{ij}^p : パス p にアーク (i, j) が $i \rightarrow j$ 向きに含まれるとき 1，そうでないとき 0 を表す定数
- O^k : 品種 k の始点
- D^k : 品種 k の終点
- X_{ij} : アーク (i, j) 上を移動するフロー量の合計を表す総フロー変数；非負の連続

変数

- x_{ij}^k : 品種 k のフローがアーク (i, j) 上を移動するフロー量を表す品種フロー変数 ; 非負の連続変数
- ξ_{ij}^s : アーク (i, j) 上の区分 s の範囲内に総フロー量が存在する場合のフロー量を表す区分フロー変数 ; 非負の連続変数
- ζ_{ij}^{ks} : アーク (i, j) 上の区分 s の範囲内に区分フロー量が存在する場合の品種 k のフロー量を表す区分品種フロー変数 ; 非負の連続変数
- z_p^k : 品種 k のフローがパス p 上を移動するフロー量を表すパスフロー変数 ; 非負の連続変数
- y_{ij}^s : アーク (i, j) の区分 s の範囲内に総フロー量が存在するとき 1 , そうでないとき 0 である区分変数 ; 0-1 変数
- π^k : 品種 k に関する需要保存式に対する双対変数 : 連続変数
- t_{ij}^k : アーク (i, j) における品種 k に関するパスフローと品種区分フローの関係式に対する双対変数 : 連続変数
- u_{ij}^s : アーク (i, j) における区分 s の品種区分フローの容量制約式に対する双対変数 : 非負の連続変数
- w_{ij}^{ks} : アーク (i, j) における区分 s , 品種 k の需要に関する強制制約式に対する双対変数 : 非負の連続変数
- r_{ij} : アーク (i, j) における区分変数の合計の上限を表す制約式に対する双対変数 : 非負の連続変数
- g_{ij}^s : アーク (i, j) における区分 s の区分変数の上限を表す制約式に対する双対変数 : 非負の連続変数

問題の定義

ノード集合 N , (S, c, f) で定義される区分的線形費用をもつ向きをもつアーク集合 A , 品種の需要 d をもつ品種集合 K が与えられている . このとき , フロー費用の合計を最小にするフロー ξ または ζ , およびフローが存在するアークの区分 y を求めよ .

6.2.2 アークフローを用いた定式化

PND に対して , Croxton ら [24] による基本の定式化 , 強い定式化および拡張した定式化による三種類の定式化を示す .

図 6.1 に示すように , アークにはフロー量に関して区分的線形となる非線形の費用関数が与えられている . 線形の区間を区分とよぶ . 費用関数は , 必ずしも区分の前後で連続である必要はないが , 下半連続であり , 各区分における線形関数の単位当りの変動費用と固定費用は非負とする .

区分 s の上限値を b_{ij}^s , 下限値を b_{ij}^{s-1} とし , $b_{ij}^0=0$ とする . アーク (i, j) の区分 s にお

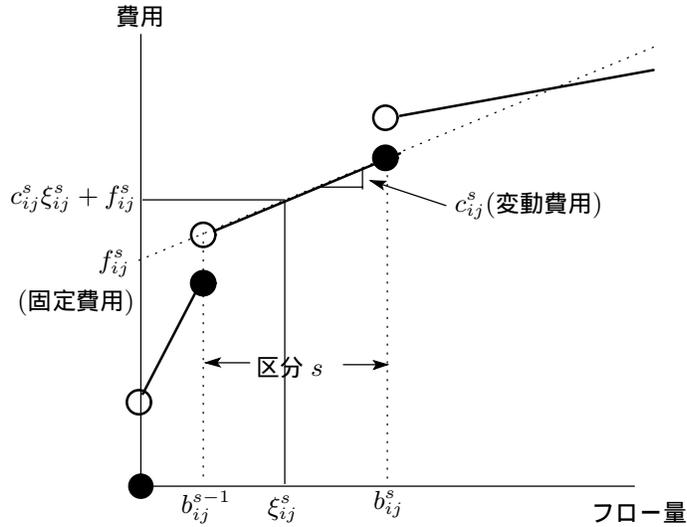


図 6.1 区分的線形関数

ける単位当りの変動費用を c_{ij}^s ，固定費用を f_{ij}^s とする．なお，フロー量が 0 であるとき，これらの費用は発生しない．

アーク (i, j) において，アーク上のフロー量を表す総フロー変数を X_{ij} ，品種 k のフロー量を表す品種フロー変数を x_{ij}^k とする．また， X_{ij} が区分 s の範囲にあるとき X_{ij} ，それ以外は 0 である区分フロー変数を ξ_{ij}^s とする．さらに， X_{ij} が区分 s の範囲にあるとき 1，それ以外は 0 を表す 0-1 変数である区分変数を y_{ij}^s とする． ξ および y は次式で表わされる．

$$\xi_{ij}^s = \begin{cases} X_{ij} & \text{if } b_{ij}^{s-1} < X_{ij} \leq b_{ij}^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_{ij}, (i, j) \in A \quad (6.1)$$

$$y_{ij}^s = \begin{cases} 1 & \text{if } b_{ij}^{s-1} < X_{ij} \leq b_{ij}^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_{ij}, (i, j) \in A \quad (6.2)$$

$y_{ij}^s = 1$ であるときには，アーク (i, j) における区分 s の区分フローが存在し，この区分フローに対するフロー費用が発生する．区分フロー量が ξ_{ij}^s であるときのフロー費用は， $b_{ij}^{s-1} < \xi_{ij}^s = X_{ij} \leq b_{ij}^s$ のとき $c_{ij}^s \xi_{ij}^s + f_{ij}^s$ ，そうでないとき 0 となるため， $c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s$ と表すことができる．

PND に対する基本的な定式化 $PAFB$ は，次のように表わされる．

(PAFB)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.3)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.4)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.5)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.6)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.7)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.8)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.9)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.10)$$

(6.3) 式は目的関数であり，変動費用と固定費用の和である総費用を最小化する．第一項は変動費用，第二項は固定費用である．(6.4) 式は，品種 k について，ノード n に入るフロー量と出るフロー量の差が，ノード n が始点であれば $-d^k$ ，終点であれば d^k ，その他のノードであれば 0 であることを表すフロー保存式である．(6.5) 式は，アーク上の品種フロー量の品種に関する合計が総フロー量に一致することを表す．(6.6) 式は，区分フロー量の区分に関する合計が総フロー量に一致することを表す．(6.7) 式は，区分変数が 1 であるとき，区分フロー量は区分の上限値と下限値の間の範囲に限定され，そうでないときは 0 となることを表す．これはアークにおける区分フローに関する強制制約式となる．(6.7) 式には区分フローの下限に等号が含まれているが，これは $y_{ij}^s = 0$ のとき， $\xi_{ij}^s = 0$ とするためである．区分フローの定義から $b_{ij}^{s-1} < \xi_{ij}^s \leq b_{ij}^s$ (または $\xi_{ij}^s = 0$) であるため，区分 s の下端のフロー量 ($\xi_{ij}^s = b_{ij}^{s-1}$) である場合は定義に反する．しかし，費用関数は下半連続であり，かつ最小化問題であるため，このような場合には最適解において区分 $s-1$ の上端のフロー量である $\xi_{ij}^{s-1} = b_{ij}^{s-1}$ を採用すればよいため，妥当な表現となる．(6.8) 式は，各アークに対して区分変数の合計が 0 または 1 であることを表し，1 をとる区分変数は高々 1 つであることを表す．(6.9) 式は品種フロー変数が非負の実数であることを表し，(6.10) 式は区分変数の 0-1 条件を表す．

基本的な定式化 PAFB は，品種フローと品種の需要に対する強制制約式を含まない弱い定式化となる．そこで，次のような強制制約式を追加する．

$$x_{ij}^k \leq d^k \sum_{s \in S_{ij}} y_{ij}^s \quad \forall (i, j) \in A, k \in K \quad (6.11)$$

(6.11) 式は，アーク (i, j) における品種 k の需要に関する品種フローの上限値を表す強制制約式である．これは，アーク (i, j) において， S_{ij} に含まれるいずれかの区分変数が 1

であれば品種 k の品種フローの上限値は d^k であり, すべての区分変数が 0 であれば品種 k のフローが存在しない, すなわち, $x_{ij}^k = 0$ であることを表す.

この強制制約式を加えた定式化を強い定式化 *PAFS* とする. *PAFS* は, 次のように表される.

(*PAFS*)

$$\text{最小化 } \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.12)$$

$$\text{条件 } \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.13)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.14)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.15)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.16)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.17)$$

$$x_{ij}^k \leq d^k \sum_{s \in S_{ij}} y_{ij}^s \quad \forall k \in K, (i, j) \in A \quad (6.18)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.19)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.20)$$

アーク (i, j) における区分 s , 品種 k に対する区分品種フロー変数を ζ_{ij}^{ks} とおく. ζ_{ij}^{ks} は, アーク (i, j) における総フロー X_{ij} が区分 s の範囲にあるときに品種 k の品種フロー x_{ij}^k となり, それ以外は 0 となる変数である.

$$\zeta_{ij}^{ks} = \begin{cases} x_{ij}^k & \text{if } b_{ij}^{s-1} < X_{ij} \leq b_{ij}^s \\ 0 & \text{otherwise} \end{cases} \quad \forall s \in S_{ij}, k \in K, (i, j) \in A \quad (6.21)$$

区分品種フロー変数と, 区分フロー変数, 品種フロー変数の関係は次式となる.

$$\xi_{ij}^s = \sum_{k \in K} \zeta_{ij}^{ks} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.22)$$

$$x_{ij}^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.23)$$

区分品種フロー変数を用いると, (6.11) 式を区分について次のように非集約化することができる.

$$\zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.24)$$

この式はアーク (i, j) における品種 k , 区分 s の需要に関する強制制約式であり, (6.11) 式よりも強い強制制約式となる.

区分品種フロー変数を用いた定式化を拡張した定式化 *PAFE* とする. *PAFE* は, 次のように表される.

(PAFE)

$$\text{最小化 } \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.25)$$

$$\text{条件 } \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.26)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.27)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.28)$$

$$\xi_{ij}^s = \sum_{k \in K} \zeta_{ij}^{ks} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.29)$$

$$x_{ij}^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.30)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.31)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.32)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.33)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.34)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.35)$$

フロー変数は非負の実数であり，区分変数は 0–1 変数であるため，三つの定式化とも組合せ最適化問題となる．0–1 変数は $\sum_{(i,j) \in A} |S_{ij}|$ 個であり，*PAFB* と *PAFS* の連続変数の数は $|A| + |A||K| + \sum_{(i,j) \in A} |S_{ij}|$ 個，*PAFE* ではさらに $|K| \sum_{(i,j) \in A} |S_{ij}|$ 個増加する．また，*PAFB* の制約式の数は $|N||K| + \sum_{(i,j) \in A} |S_{ij}| + 3|A|$ 本，*PAFS* では $|A||K|$ 本増加し，*PAFE* ではさらに $\sum_{(i,j) \in A} |S_{ij}| + |K| \sum_{(i,j) \in A} |S_{ij}|$ 本増加する．このように，*UND* や *CND* に比べて，これらの定式化は大規模なものとなり，汎用の数理最適化ソフトウェアを用いて直接的に最適解を求めることは困難となる．

6.2.3 パスフローを用いた定式化

パス上のフロー量を表すパスフロー変数を用いた拡張した定式化を *PPFE* とする．*PPFE* は，次のようになる．

(PPFE)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.36)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (6.37)$$

$$x_{ij}^k = \sum_{p \in P^k} \delta_{ij}^p z_p^k \quad \forall k \in K, (i, j) \in A \quad (6.38)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.39)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.40)$$

$$\xi_{ij}^s = \sum_{k \in K} \zeta_{ij}^{ks} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.41)$$

$$x_{ij}^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.42)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.43)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.44)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.45)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (6.46)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.47)$$

(6.37) 式は，品種 k のパスフローの合計が品種 k の需要 d^k に一致することを表す需要保存式である．(6.38) 式は，パスフローと品種フローの関係式である．また，(6.46) 式は，パスフロー変数の非負制約である．

(6.38) ~ (6.42) 式を用いて， X ， x ， ξ を整理すると，パスフロー変数を用いた拡張した定式化 $PPFE$ は次のような y ， z ， ζ を変数とする問題 $PPFE'$ にまとめることができる．

(PPFE')

$$\text{最小化 } \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.48)$$

$$\text{条件 } \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (6.49)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.50)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.51)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.52)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.53)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (6.54)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.55)$$

パスフロー変数の数は膨大であるため、アークフロー変数による定式化よりもさらに大規模な組合せ最適化問題となる。しかし、最適解に含まれるパスフロー変数は比較的少ないため、必要なパスフロー変数だけを列挙する列生成法を用いると、問題を効率的に解くことができる。

6.2.4 容量制約と多重ダミーアーク

区分数が1である区分的線形費用をもつ問題を考えると、この問題は区分の上限值が容量である容量制約をもつネットワーク設計問題と見なすことができる。そこで、Kim-Pardalos[50]と同様に、区分的線形費用をもつ問題に対して、アークにおける各区分を一本のダミーアークと見なし、図6.2に示すように、一本のアークに対して多重のダミーアークを設定する。対応する区分の上限値をダミーアークの容量とし、デザイン費用とフロー費用は対応する区分の固定費用と変動費用とする。

アークを多重のダミーアークに置き換えたとき、定式化では(6.7)式を次式で置き換えれば良い。

$$0 \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.56)$$

区分変数 y_{ij}^s はアーク (i, j) のダミーアーク s に対する0-1であるデザイン変数となる。

PAFBにおいて、(6.16)式を(6.56)式で置き換えた問題をPAFBDとおく。PAFBDは最小化問題であることから、この問題の最適解では、各ノード間の多重アークの中で、フロー量に対して費用が最小となるダミーアークにフローが移動することになる。一方、PAFBにおいても、最適解において、フロー量に対して費用が最小となる区分上にフローが存在する。これは、PAFBDがPAFBと等価な問題となることを意味する。このように、PAFBDはアークが容量をもつ問題に帰着できるので、CNDに対する容量スケールリング法を適用することができる。

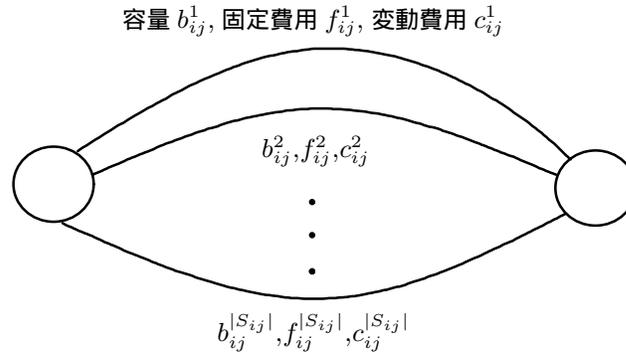


図 6.2 多重ダミーアーク

同様に $PAFS$ において, アークを多重アークに置き換え, (6.16) 式を (6.56) 式とした問題を $PAFSD$ とする. また, 同様に $PAFE$ において, (6.31) 式を (6.56) 式で置き換えた問題を $PAFED$ とする. $PAFSD$ および $PAFED$ に対しても, CND に対する容量スケールリング法を適用することができる.

さらに, $PPFE'$ に対して, アークを多重アークに置き換え, (6.51) 式を次式で置き換えた問題を $PPFED$ とする.

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.57)$$

$PPFED$ に対しても, CND に対する容量スケールリング法を適用することができる.

6.2.5 フロー変数に関する被約費用と双対問題

$PPFED$ において, 0-1 である区分変数を 0 から 1 の連続数に緩和した線形緩和問題 $PPFEL$ を考える.

(PPFEL)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.58)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (6.59)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i,j) \in A \quad (6.60)$$

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.61)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i,j) \in A \quad (6.62)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i,j) \in A \quad (6.63)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (6.64)$$

$$0 \leq y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.65)$$

PPFEL におけるフロー変数 z および ζ に関する被約費用を求める。(6.59), (6.60) 式に対する双対変数を π, t とし, (6.61) の右側の式, (6.62) 式の右側の式に対する非負の双対変数を u, w とし, これらを用いて, z および ζ に関する PPFEL に対する Lagrange 双対関数 $LF(z, \zeta)$ を作成する.

$$\begin{aligned} LF(z, \zeta) &= \sum_{(i,j) \in A} \sum_{s \in S_{ij}} c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} - \sum_{k \in K} \sum_{p \in P^k} z_p^k \pi^k + \sum_{(i,j) \in A} \sum_{k \in K} \left(\sum_{p \in P^k} \delta_{ij}^p z_p^k - \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \right) t_{ij}^k \\ &\quad + \sum_{(i,j) \in A} \sum_{s \in S_{ij}} \sum_{k \in K} \zeta_{ij}^{ks} u_{ij}^s + \sum_{(i,j) \in A} \sum_{s \in S_{ij}} \sum_{k \in K} \zeta_{ij}^{ks} w_{ij}^{ks} \\ &= \sum_{(i,j) \in A} \sum_{s \in S_{ij}} \sum_{k \in K} (c_{ij}^s + u_{ij}^s + w_{ij}^{ks} - t_{ij}^k) \zeta_{ij}^{ks} + \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{(i,j) \in A} \delta_{ij}^p t_{ij}^k - \pi^k \right) z_p^k \end{aligned} \quad (6.66)$$

したがって, ζ に対する被約費用 r および z に対する被約費用 q は, 次のようになる.

$$r_k^s = c_{ij}^s + u_{ij}^s + w_{ij}^{ks} - t_{ij}^k \quad \forall k \in K, s \in S_{ij}, (i,j) \in A \quad (6.67)$$

$$q_{ij}^{ks} = \sum_{(i,j) \in A} \delta_{ij}^p t_{ij}^k - \pi^k \quad \forall p \in P^k, k \in K \quad (6.68)$$

(6.63) 式に対する非負の双対変数を r とし, (6.65) の右側の式に対する非負の双対変数を g とする. このとき, PPFEL の双対問題 DU は次のようになる.

(DU)

$$\text{最大化} \quad \sum_{k \in K} d^k \pi^k - \sum_{(i,j) \in A} \left(r_{ij} + \sum_{s \in S_{ij}} g_{ij}^s \right) \quad (6.69)$$

$$\text{条件} \quad \sum_{(i,j) \in A} \delta_{ij} t_{ij}^k \geq \pi^k \quad \forall p \in P^k, k \in K \quad (6.70)$$

$$t_{ij}^k \leq c_{ij}^k + u_{ij}^s + w_{ij}^{ks} \quad \forall k \in K, s \in S_{ij}, (i,j) \in A \quad (6.71)$$

$$b_{ij}^s u_{ij}^s + \sum_{k \in K} d^k w_{ij}^{ks} \leq f_{ij}^s + r_{ij} + g_{ij}^s \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.72)$$

$$u_{ij}^s \geq 0 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.73)$$

$$w_{ij}^{ks} \geq 0 \quad \forall k \in K, s \in S_{ij}, (i,j) \in A \quad (6.74)$$

$$r_{ij} \geq 0 \quad \forall (i,j) \in A \quad (6.75)$$

$$g_{ij}^s \geq 0 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.76)$$

6.3 アークフローを用いた定式化に対する容量スケールリング法

6.3.1 容量スケールリング法

アークの容量制約をもつ問題において、なんらかの手段を用いて問題の最適フローを求めることができれば、容量をこのフロー量に変更しても最適値は変わらない。さらに、このように容量を変更した問題の線形緩和問題を考える。この線形緩和問題においても最適フローは実行可能解となることから、緩和問題の解においてフロー量と容量が一致し、0-1変数が1となることが期待できる。ただし、フロー量が0である場合、0-1変数は0と考える。

しかし、最適フローを求めること自体が問題の目的であるため、直接、最適フローを求めることはできない。そこで、逆に、線形緩和問題において、容量を少しずつ変更しながら問題を解き直し、容量が最適フロー量に近づくように容量を設定することができれば、元の問題の最適フローに近いフローが見つかることが期待できる。容量スケールリング法は、このように線形緩和問題のフロー解をもとに容量を変更して繰り返し線形緩和問題を解き、0-1変数解を導く近似解法である。

6.3.2 基本的な定式化に対する容量スケールリング法

はじめに、基本的な定式化である *PAFB* に対する容量スケールリング法を示す。*PAFBD* において、0-1変数であるダミーアークに対応する区分変数 y を 0 から 1 の連続数に緩和した線形緩和問題 *PAFBL* を考える。

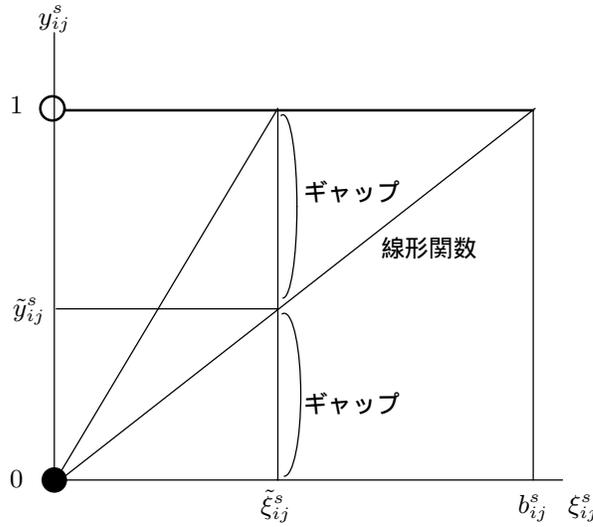


図 6.3 線形関数とギャップ

(PAFBL)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.77)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.78)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.79)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.80)$$

$$0 \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.81)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.82)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.83)$$

$$0 \leq y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.84)$$

この PAFBL は最小化問題であり、 y_{ij}^s の下限に関する制約式は (6.81) 式のみである。このため、最適解において、区分フロー変数と区分変数の関係は、(6.81) 式から $y_{ij}^s = \xi_{ij}^s / b_{ij}^s$ となり、図 6.3 に示すように、 y_{ij}^s は ξ_{ij}^s の線形関数で表わすことができる。

PAFBL における区分フロー解を $\tilde{\xi}_{ij}^s$ としたとき、対応する区分変数解は $\tilde{y}_{ij}^s = \tilde{\xi}_{ij}^s / b_{ij}^s$ となる。 y_{ij}^s は本来 0 または 1 であるため、図 6.3 に示すように \tilde{y}_{ij}^s が小数値を取った場合には、これらの間にギャップが発生する。そこで、区分フローが変化しないものと想定すると、アーク容量 b_{ij}^s を $\tilde{\xi}_{ij}^s$ に変更すれば $\tilde{y}_{ij}^s = 1$ となり、このギャップが埋まることが期待できる。

実際には、アーク容量を変更させた問題を解き直すと、区分フロー自体が変化してし

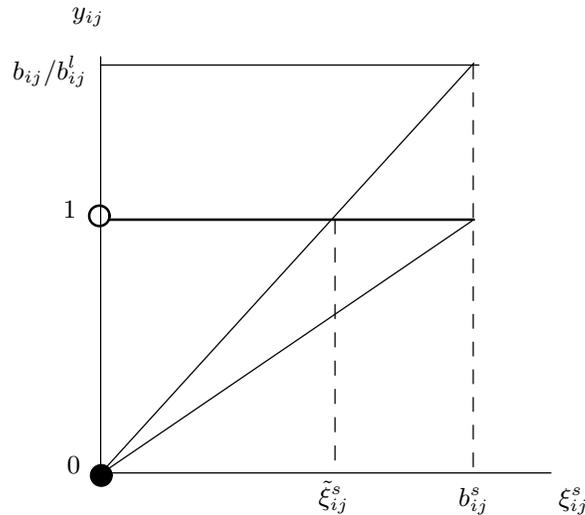


図 6.4 区分変数とフロー変数の関係

まい，ギャップが埋まる保証はない．そこで，パラメータ λ を用いて，アーク容量 b_{ij}^s を徐々に変更した問題を解き直し，新たな解における区分フローが大きく変化しないように制御する．区分変数解が 0 または 1 に収束するまで，この操作を繰り返す．

ダミーアークの l 回目の容量を b^l とおき，初期値 b^1 を b とする． $l-1$ 回目の繰り返しにおいて，容量を b^{l-1} とした線形緩和問題を解き直し，新たな区分フロー解 $\tilde{\xi}$ を求める．続いて，パラメータ $\lambda (0 < \lambda \leq 1)$ を用いて， b^l を次のように更新する．

$$b_{ij}^{sl} := \lambda \tilde{\xi}_{ij}^s + (1 - \lambda) b_{ij}^{s^{l-1}} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.85)$$

この式は，アーク (i, j) ，区分 s に対応する区分フロー量 $\tilde{\xi}_{ij}^s$ と $l-1$ 回目の容量 $b_{ij}^{s^{l-1}}$ の凸結合によって， l 回目の容量 b_{ij}^{sl} を定めることを意味する． λ によって容量の急激な変更を防ぎ，問題を解き直して得られるフローの大きな変化を防止する．

一方， $\tilde{\xi}_{ij}^s = b_{ij}^{s^{l-1}} \tilde{y}_{ij}^s$ の関係から，(6.85) 式は次式としても表現できる．

$$b_{ij}^{sl} := \lambda b_{ij}^{s^{l-1}} \tilde{y}_{ij}^s + (1 - \lambda) b_{ij}^{s^{l-1}} = \{1 - \lambda(1 - \tilde{y}_{ij}^s)\} b_{ij}^{s^{l-1}} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.86)$$

この式は， $\tilde{y}_{ij}^s < 1$ の場合では，容量を $\lambda(1 - \tilde{y}_{ij}^s)$ の比率で減少させることを意味する．また， $\tilde{y}_{ij}^s > 1$ の場合では， $\lambda(\tilde{y}_{ij}^s - 1)$ の比率で増加させることを意味する．

容量の変更を行っているが， ξ_{ij}^s は本来のアーク容量である b_{ij}^s 以下であればよい．そこで，図 6.4 に示すように， y_{ij}^s の上限は b_{ij}^s / b_{ij}^{sl} となるために，(6.84) 式の y_{ij}^s の係数を b_{ij}^{sl} / b_{ij}^s に変更する．このとき， y_{ij}^s は 1 以上の値を取ることができるが，区分フロー ξ_{ij}^s は本来の容量 b_{ij}^s 以下であることが保証される．なお，この変更を (6.82) 式にも反映し，(6.82) 式における y_{ij}^s の係数を b_{ij}^{sl} / b_{ij}^s とする．

このように，容量スケール法では，変更した容量以上のフローを認める．このため，容量 b_{ij}^{sl} は，厳密な意味での容量ではなく， y_{ij}^s の取り得る範囲を変更させることによ

り，線形緩和におけるデザイン費用を変化させるパラメータ的な機能を果たしていることになる．

容量を b^l とした l 回目の繰り返しにおける $PAFBL$ を $PAFBL_l$ とする．

($PAFBL_l$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.87)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.88)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.89)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.90)$$

$$0 \leq \xi_{ij}^s \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.91)$$

$$\sum_{s \in S_{ij}} \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.92)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.93)$$

$$0 \leq \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.94)$$

この問題は線形計画問題であるため，汎用の数理最適化ソフトウェアを用いて解くことができる．

容量スケールリング法では，繰り返し回数とともに多くのデザイン変数は 0 または 1 に収束する．しかしながら，すべての 0-1 変数が 0 または 1 に収束する保証はないため，繰り返し回数の上限 l_{max} を設定する． $PAFB$ に対する容量スケールリング法を示す．

$PAFB$ に対する容量スケールリング法

[ステップ 1] スケールリングパラメータを λ ，繰り返し回数の上限を l_{max} とする．

$b^l := b, l := 0$ とする．

[ステップ 2] $l := l + 1$ とする． $PAFBL_l$ を解き， $\tilde{y}, \tilde{\xi}$ を求める．

[ステップ 3] $b^l := \lambda \tilde{\xi} + (1 - \lambda) b^{l-1}$ ，または $b^l := \{1 - \lambda(1 - \tilde{y})\} b^{l-1}$ とする．

[ステップ 4] \tilde{y} の成分がすべて 0 または 1 であるか， $l = l_{max}$ であれば終了，そうでなければ，ステップ 2 へ戻る．

6.3.3 強い定式化および拡張した定式化に対する容量スケールリング法

$PAFB$ と同様に，強い定式化 $PAFS$ に対して，多重アークを用い， l 回目のアーク容量を b^l とした問題の線形緩和問題を $PAFSL_l$ とする．

(PAFSL_l)

$$\text{最小化 } \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.95)$$

$$\text{条件 } \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.96)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.97)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.98)$$

$$0 \leq \xi_{ij}^s \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.99)$$

$$\sum_{s \in S_{ij}} \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.100)$$

$$x_{ij}^k \leq d^k \sum_{s \in S_{ij}} y_{ij}^s \quad \forall k \in K, (i, j) \in A \quad (6.101)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.102)$$

$$0 \leq \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.103)$$

同様に，拡張した定式化 *PAFE* に対して，多重アークを用い，容量を b^l とした問題の線形緩和問題を *PAFEL_l* とする．

(PAFEL_l)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.104)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.105)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.106)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.107)$$

$$\xi_{ij}^s = \sum_{k \in K} \zeta_{ij}^{ks} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.108)$$

$$x_{ij}^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.109)$$

$$0 \leq \xi_{ij}^s \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.110)$$

$$\sum_{s \in S_{ij}} \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.111)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.112)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.113)$$

$$0 \leq \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.114)$$

PAFSL_l および PAFEL_l は線形計画問題であるため、汎用の数理最適化ソフトウェアを用いて解くことができる。

PAFS と PAFE に対する容量スケールリング法は、PAFBL_l のかわりに PAFSL_l と PAFEL_l を用いる。PAFB と同様に、 $l-1$ 回目の繰り返しにおいて、容量を b^{l-1} とした線形緩和問題を解き直し、新たな区分フロー $\tilde{\xi}$ を求める。続いて、パラメータ $\lambda (0 < \lambda \leq 1)$ を用いて、 b^l を次のように更新する。

$$b_{ij}^{sl} := \lambda \tilde{\xi}_{ij}^s + (1 - \lambda) b_{ij}^{sl-1} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.115)$$

一方、PAFSL_l および PAFEL_l において、デザイン変数の下限である (6.101) 式や (6.112) 式があるために、必ずしも (6.99) 式の右側の式および (6.110) 式の右側の式が等式で成り立つとは限らず、 $b_{ij}^{sl} y_{ij}^s$ は ξ_{ij} の上限値となる。そこで、容量の変更は ξ_{ij} ではなく、 ξ_{ij} の上限値である $b_{ij}^{sl} y_{ij}^s$ を用いることもできる。このとき、 b^l は次のように更新することもできる。

$$b_{ij}^{sl} := \lambda b_{ij}^{sl-1} \tilde{y}_{ij}^s + (1 - \lambda) b_{ij}^{sl-1} = \{1 - \lambda(1 - \tilde{y}_{ij}^s)\} b_{ij}^{sl-1} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.116)$$

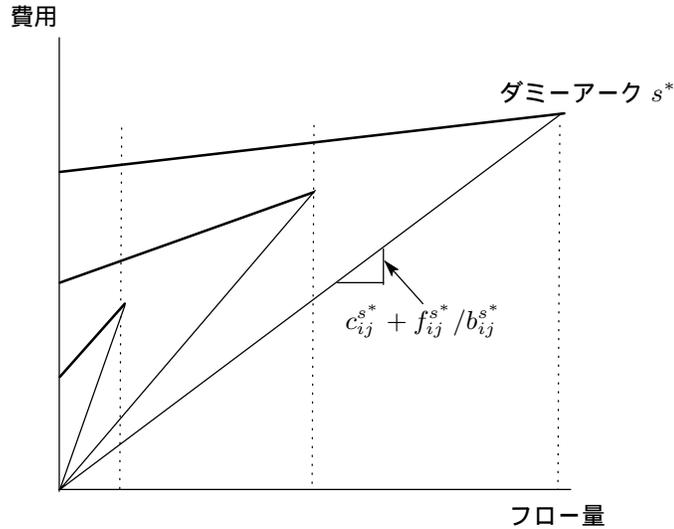


図 6.5 ダミーアークとフロー量

6.3.4 近似解法

デザイン変数の固定法

$PAFB$ に対する容量スケールリング法では、容量 b を変更して、 $PAFBL_l$ を繰り返し解き直す。 $PAFBL_l$ は最小化問題であることから、最適解において $y = \xi/b^l$ が成り立つので、目的関数値は

$$\sum_{(i,j) \in A} \sum_{s \in S_{ij}} \left(c_{ij}^s + \frac{f_{ij}^s}{b_{ij}^{sl}} \right) \xi_{ij}^s \quad (6.117)$$

と等価になる。このため、図 6.5 に示すように、最適解では多重のダミーアークの中で傾きである $c_{ij}^s + f_{ij}^s/b_{ij}^{sl}$ が最小であるダミーアーク s^* にフローが集中する。単純に線形緩和解が正である区分変数を 1 としてしまうと、 $PAFB$ において総フロー量と区分変数の範囲が対応しない可能性が生じる。そこで、 $PAFBL_l$ の総フローの線形緩和解 \tilde{X}_{ij} がどの区分の範囲にあるかによって、区分変数の値を設定する。図 6.6 に示すように、 \tilde{X}_{ij} が上限 b_{ij}^s と下限 b_{ij}^{s-1} の範囲にある区分に対応する区分変数を 1、それ以外を 0 とした区分変数を \bar{y}_{ij}^s とする。

$$\bar{y}_{ij}^s = \begin{cases} 1 & \text{if } b_{ij}^{s-1} < \tilde{X}_{ij} \leq b_{ij}^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_{ij}, (i,j) \in A \quad (6.118)$$

このとき、 \bar{y} は $PAFB$ の制約式である (6.7) 式と (6.8) 式を満足する。

次に、区分変数を \bar{y} としたネットワーク上で、次のような上下限付きの多品種フロー問題 $MCFLU(\bar{y})$ を作成する。

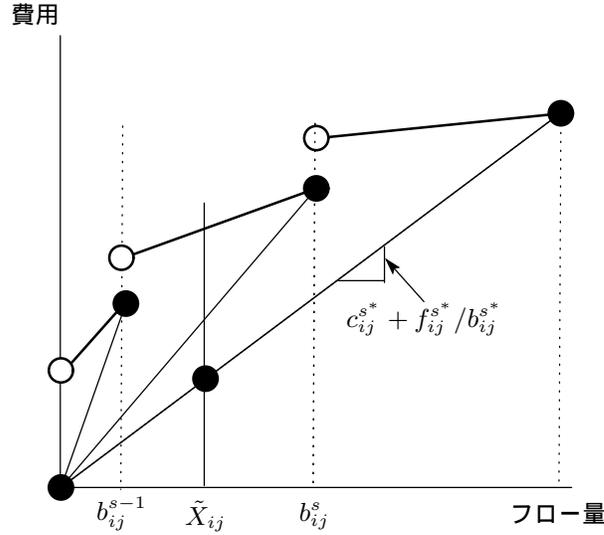


図 6.6 区分フロー量と区分変数値

$(MCFLU(\bar{y}))$

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s \bar{y}_{ij}^s) \quad (6.119)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.120)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.121)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.122)$$

$$\begin{cases} b_{ij}^{s-1} < \xi_{ij}^s \leq b_{ij}^s & \text{if } \bar{y}_{ij}^s = 1 \\ \xi_{ij}^s = 0 & \text{if } \bar{y}_{ij}^s = 0 \end{cases} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.123)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.124)$$

ここで、(6.123) 式は区分フローの範囲を限定する制約である。

$MCFLU(\bar{y})$ は線形計画問題であるため、汎用の数理最適化ソフトウェアを用いて解くことができる。 $MCFLU(\bar{y})$ の実行可能なフロー解を求めることができれば、 \bar{y} およびフロー解は $PAFB$ の実行可能解となり、 $MCFLU(\bar{y})$ の目的関数値は PND の上界値となる。

一方、 $PAFS$ および $PAFE$ では、区分変数に関する強制制約式 (6.101) や (6.112) が含まれている。このため、緩和策において、多重アークの中で複数のダミーアークのフロー量が正となり、複数の区分変数が正になる可能性がある。しかし、 $PAFB$ の場合と同様に、総フロー変数の線形緩和策を用いて、(6.118) 式を用いて区分変数を設定すれば、制約式を満足する区分変数を一意に設定することができる。

限定した分枝限定法

容量スケールリング法では、繰り返し回数とともに、多くの 0-1 変数である区分変数が 0 または 1 に収束する傾向がある。汎用の数理最適化ソフトウェアの分枝限定法を用いると、少数の 0-1 変数をもつ計画問題に対しては比較的短時間で最適解を求めることができる。そこで、 l 回目の繰り返しにおいて、制限を付けた分枝限定法を行う。この制限は、次のようなものである。PAFBL $_l$ の区分変数解を \tilde{y} とし、0 または 1 に収束していない区分変数の合計を ω_l 個とする。 ω_l がある決められた数である ψ 個以下となった繰り返し回数において、各アークに関して、 $\sum_{s \in S_{ij}} \tilde{y}_{ij}^s$ が 0 に近ければこれらのアークを選択しないものとする。

$$\sum_{s \in S_{ij}} y_{ij}^s = 0 \quad \text{if} \quad \sum_{s \in S_{ij}} \tilde{y}_{ij}^s \leq \epsilon_0 \quad \forall (i, j) \in A \quad (6.125)$$

ここで、 ϵ_0 は $0 < \epsilon_0 < 1$ である定数である。

PAFB における区分変数に制限をつけた問題 PAFBR(\tilde{y}) は次のようになる。

(PAFBR(\tilde{y}))

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s y_{ij}^s) \quad (6.126)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.127)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.128)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.129)$$

$$b_{ij}^{s-1} y_{ij}^s \leq \xi_{ij}^s \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.130)$$

$$\begin{cases} \sum_{s \in S_{ij}} y_{ij}^s = 0 & \text{if } \sum_{s \in S_{ij}} \tilde{y}_{ij}^s \leq \epsilon_0 \\ \sum_{s \in S_{ij}} y_{ij}^s \leq 1 & \text{otherwise} \end{cases} \quad (i, j) \in A \quad (6.131)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.132)$$

$$y_{ij}^s \in \{0, 1\} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.133)$$

PAFBR(\tilde{y}) に対して、分枝限定法を適用すれば、PAFB の近似解と上界値を求めることができる。

しかし、このように 0-1 変数を限定した問題であっても、 ψ の大きさ、問題の規模や性質によって、分枝限定法により最適解を求めることが困難な場合がある。そのため、分枝限定法は一定時間で打ち切ることにし、その時点で得られた最良の解を近似解とする。また、 ψ を一定値とした場合、連続した繰り返し回数において分枝限定法を繰り返し、同一の解を生成することが多くなる。そこで、分枝限定法を適用する毎に ψ を β だけ減少させ、解の多様性を確保する。

一方, $PAFS$ および $PAFE$ では, これらの定式化に (6.125) 式を加えた $PAFSR(\tilde{y})$ および $PAFER(\tilde{y})$ を解くことになる.

6.3.5 アークフローによる定式化に対する容量スケールリング法

二種類の近似解法を含めた $PAFB$ に対する容量スケールリング法を示す.

$PAFB$ に対する容量スケールリング法

- [ステップ 1] 最良の上界値を UB , 繰り返し回数の上限を l_{max} , スケールリングパラメータを λ , 閾値を ϵ_0, ϵ_1 および ϵ_2 , 分枝限定法基準を ψ , その変化量を β とする. $b^l := b, UB := \infty, l := 0$ とする.
- [ステップ 2] $l := l + 1$ とする. $PAFBL_l$ を解き, 最適解 $\tilde{y}, \tilde{\xi}$ を求める.
- [ステップ 3] $b^l := \lambda \tilde{\xi} + (1 - \lambda)b^{l-1}$ または $b^l := \{1 - \lambda(1 - \tilde{y})\}b^{l-1}$ とする.
- [ステップ 4] \tilde{y} を求め, $MCFLU(\tilde{y})$ を解き上界値 UB_l を求める. $UB_l \leq UB$ であれば, $UB := UB_l$ とする.
- [ステップ 5] $\epsilon_1 < \tilde{y}_{ij}^s < \epsilon_2 (s \in S_{ij}, (i, j) \in A)$ である区分変数の数を ω_l とする. $\omega_l \leq \psi$ であれば, ϵ_0 を用いて $PAFBR(\tilde{y})$ を作成し, 分枝限定法を適用して $PAFBR(\tilde{y})$ の上界値 UB_R を求める. $UB_R \leq UB$ であれば, $UB := UB_R$ とする. $\psi := \psi - \beta$ とする.
- [ステップ 6] $l = l_{max}$ であれば終了, そうでなければステップ 2 へ戻る.

次に, 二種類の近似解法を含めた $PAFS$ および $PAFE$ に対する容量スケールリング法を示す.

$PAFS/PAFE$ に対する容量スケールリング法

- [ステップ 1] 最良の上界値を UB , 繰り返し回数の上限を l_{max} , スケールリングパラメータを λ , 閾値を ϵ_0, ϵ_1 および ϵ_2 , 分枝限定法基準を ψ , その変化量を β とする. $b^l := b, UB := \infty, l := 0$ とする.
- [ステップ 2] $l := l + 1$ とする. $PAFSL_l$ または $PAFEL_l$ を解き, 最適解 $\tilde{y}, \tilde{\xi}$ および \tilde{X} を求める.
- [ステップ 3] $b^l := \lambda \tilde{\xi} + (1 - \lambda)b^{l-1}$ または $b^l := \{1 - \lambda(1 - \tilde{y})\}b^{l-1}$ とする.
- [ステップ 4] \tilde{y} を求め, $MCFLU(\tilde{y})$ を解き上界値 UB_l を求める. $UB_l \leq UB$ であれば, $UB := UB_l$ とする.
- [ステップ 5] $\epsilon_1 < \tilde{y}_{ij}^s < \epsilon_2 (s \in S_{ij}, (i, j) \in A)$ である区分変数の数を ω_l とする. $\omega_l \leq \psi$ であれば, ϵ_0 を用いて $PAFSR(\tilde{y})$ または $PAFER(\tilde{y})$ を作成し, 分枝限定法を適用して上界値 UB_R を求める. $UB_R \leq UB$ であれば, $UB := UB_R$ とする. $\psi := \psi - \beta$ とする.
- [ステップ 6] $l = l_{max}$ であれば終了, そうでなければステップ 2 へ戻る.

6.4 パスフローを用いた定式化に対する容量スケールリング法

6.4.1 容量スケールリング

$PPFED$ はアークが容量 b をもつ問題となるので, CND に対する容量スケールリング法が適用できる. $PPFED$ において, 0-1 変数である区分変数を 0 から 1 の連続数に緩和した線形問題 $PPFEL$ を作成する.

($PPFEL$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.134)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (6.135)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.136)$$

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^s y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.137)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.138)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.139)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (6.140)$$

$$0 \leq y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.141)$$

$PPFEL$ の最適解における区分フローを $\tilde{\xi} = (\tilde{\xi}_{ij}^s) = (\sum_{k \in K} \tilde{\zeta}_{ij}^{ks})$, 区分変数値を \tilde{y} とする. y_{ij}^s は本来は 0 または 1 であるが, \tilde{y}_{ij}^s の多くは小数値を取る可能性があるため, これらの間にギャップが存在する可能性がある. そこで, 区分フローが変化しないものと想定し, アーク容量 b_{ij}^s を $\tilde{\xi}_{ij}^s$ に変更すれば, $\tilde{y}_{ij}^s = 1$ となり, このギャップを埋めることが期待できる.

実際には, 容量を変更させた問題を解き直すと, フロー自体が変化してしまい, 必ずしもギャップが埋まる保証はない. そこで, パラメータ λ を用いて, 容量 b を徐々に変更した問題を解き直し, 求められたフローが大きく変化しないように制御する.

l 回目の繰り返しにおいて, アーク容量を b^l とした線形緩和問題を $PPFEL_l$ とする.

(PPFEL_l)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.142)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (6.143)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.144)$$

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.145)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.146)$$

$$\sum_{s \in S_{ij}} \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.147)$$

$$0 \leq \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.148)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (6.149)$$

ここで, ξ_{ij}^s は本来の容量 b_{ij}^s 以下であればよいので, (6.141) 式の y_{ij}^s の係数を変更し, $0 \leq b_{ij}^{sl}/b_{ij}^s y_{ij}^s \leq 1$ とする. なお, この変更を (6.139) 式にも反映し, y_{ij}^s の係数を b_{ij}^{sl}/b_{ij}^s とする.

アーク (i, j) , 区分 s に対応するダミーアークの l 回目の繰り返しにおけるアーク容量を b^l とおき, b^l の初期値を b とする. 容量を b^{l-1} とした PPFEL_{l-1} を解き直し, 区分フロー $\tilde{\xi}$ と区分変数 \tilde{y} を求める. 続いて, パラメータ $\lambda (0 \leq \lambda \leq 1)$ を用いて, b^l を次のように更新する.

$$b_{ij}^{sl} := \lambda \tilde{\xi}_{ij}^s + (1 - \lambda) b_{ij}^{sl-1} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.150)$$

λ によって, b^l の急激な変更を防ぎ, フローの大きな変化を防止する.

一方, PPFEL_l では, デザイン変数の下限である (6.146) 式の右側の式があるために, (6.145) 式が等式で成り立つとは限らず, $b_{ij}^{sl} y_{ij}^s$ は ξ_{ij}^s の上限値となる. そこで, 容量の変更は ξ_{ij}^s ではなく, ξ_{ij}^s の上限値である $b_{ij}^{sl} y_{ij}^s$ を用いることもできる.

$$b_{ij}^{sl} := \lambda b_{ij}^{sl-1} \tilde{y}_{ij}^s + (1 - \lambda) b_{ij}^{sl-1} = \{1 - \lambda(1 - \tilde{y}_{ij}^s)\} b_{ij}^{sl-1} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.151)$$

6.4.2 限定主問題

PPFEL_l において, 取り得るパスの数は非常に多いため, あらかじめすべてのパスを列挙した問題を解くことは困難である. そこで, はじめに, 適当なパスの部分集合からなる問題を作成する. このように, 変数を制限した問題を限定主問題とよぶ. 続いて, 逐次, 基底に入るべき変数を生成し, これらを問題に加えて解き直す. PPFEL_l において, 生成する変数は z と ζ である. 生成する変数が単体法の列に相当することから, このよ

うな方法を列生成法とよぶ．また，同時に必要な制約式を生成する．生成する制約式は単体法の行に相当することから，このような方法を行生成法とよぶ．

$PPFEL_l$ において，品種 $k(\in K)$ の適当なパスの部分集合 \bar{P}^k が求められているものとする． $\bar{P} = (\bar{P}^k)$ を用いると，限定主問題 $PPFELR_l(\bar{P})$ は次のように表わされる．

$(PPFELR_l(\bar{P}))$

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.152)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = d^k \quad \forall k \in K \quad (6.153)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall (i, j) \in A_{\bar{P}^k}, k \in K \quad (6.154)$$

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.155)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall s \in S_{ij}, (i, j) \in A_{\bar{P}^k}, k \in K \quad (6.156)$$

$$\sum_{s \in S_{ij}} \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall (i, j) \in A \quad (6.157)$$

$$0 \leq \frac{b_{ij}^{sl}}{b_{ij}^s} y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.158)$$

$$z_p^k \geq 0 \quad \forall p \in \bar{P}^k, k \in K \quad (6.159)$$

ここで， $A_{\bar{P}^k}$ は \bar{P}^k に含まれるパスが含むアークの集合である．このため，(6.154) 式と (6.156) 式はアーク (i, j) を通過する品種 k のパスフロー変数が生成されているときのみ存在する制約式となる．

$PPFELR_l(\bar{P})$ は線形計画問題であるため，パスの部分集合の要素数が比較的少なければ，汎用の数値最適化ソフトウェアを用いて比較的容易に解くことができる．

6.4.3 列生成法と行生成法

$PPFELR_l(\bar{P})$ は，変数が限定された問題である．このため， $PPFEL_l$ の最適解を求めるためには，被約費用が負である基底に入るべき変数を生成しなければならない．そこで， $PPFELR_l(\bar{P})$ の最適双対変数値を用いた価格付け問題を解き，被約費用が負である変数を求める必要がある．このような変数を生成し，対応するパスを \bar{P}^k に加え，パスフロー変数を問題に加え， $A_{\bar{P}^k}$ を更新し，再度 $PPFELR_l(\bar{P})$ を解き直す．被約費用が負である変数が存在しなくなるまで，この操作を繰り返す．被約費用が負である変数が存在しなければ， $PPFEL_l$ の最適解が得られたことになる．

双対変数 $\pi, t, u(\geq 0), w(\geq 0)$ を用いると， z に対する被約費用 r ，および ζ に対

する被約費用 q は次のようになる .

$$r_p^k = \sum_{(i,j) \in A} \delta_{ij}^p t_{ij}^k - \pi^k \quad \forall p \in P^k, k \in K \quad (6.160)$$

$$q_{ij}^{ks} = c_{ij}^s + u_{ij}^s + w_{ij}^{ks} - t_{ij}^k \quad \forall k \in K, s \in S_{ij}, (i, j) \in A \quad (6.161)$$

z と ζ の間には ,

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall k \in K, (i, j) \in A \quad (6.162)$$

の関係がある . ある変数 z_p^k が基底に入り正の値をとると , (6.162) の左辺の値も正となる . そのため , いくつかの ζ_{ij}^{ks} が基底に入ることになる . そこで , 被約費用が負であるパスフロー変数 z_p^k を生成する . 続いて , z_p^k に対応するまだ生成されていない ζ_{ij}^{ks} を生成する .

t_{ij}^k に対応する (6.162) が生成されていない , すなわち (6.162) 式に含まれる z_p^k , $\zeta_{ij}^{ks} (s \in S_{ij})$ が生成されていない場合は , t_{ij}^k の値が定義されていない . そこで , このような場合を考えて , 双対実行可能となるように t_{ij}^k の値を設定する . このとき , (6.162) の両辺の変数値が 0 で等号が成り立つと考え , t_{ij}^k に非負の値を設定する . ここで , t は z および ζ の両方の被約費用に含まれていることに注意する . z_p^k の被約費用に含まれる $\sum_{(i,j) \in A} \delta_{ij}^p t_{ij}^k$ は , アークの長さを t_{ij}^k としたときのパス p の長さであり , π^k は品種 k の始点・終点間の現在の最短距離である . 一方 , π^k は双対問題 DU の目的関数に含まれ , 非負の係数を持ち , かつ DU は最大化問題である . このため , 最適双対解では π^k の値は条件内で最大値をとり , このため t_{ij}^k の値も条件内で最大値をとる .

そこで , $\zeta_{ij}^{ks} (s \in S_{ij})$ が非基底変数である範囲内で , まだ生成されていない変数 t_{ij}^k の値が最大となるように設定する . すなわち , アーク (i, j) , 品種 k において , 区分 $s (s \in S_{ij})$ に関する ζ_{ij}^{ks} の被約費用が最小である区分の被約費用が 0 となるように , まだ生成されていない t_{ij}^k の値を設定する . なお , 関係する w_{ij}^{ks} は生成されていないため , 0 と考える . このとき , t_{ij}^k は次のようになる .

$$\bar{t}_{ij}^k = \min_{s \in S_{ij}} \{c_{ij}^s + u_{ij}^s + w_{ij}^{ks}\} \quad \forall (i, j) \in A \setminus A_{\bar{P}^k}, k \in K \quad (6.163)$$

ここで , $A \setminus A_{\bar{P}^k}$ は品種 k のパスに含まれていないアークの集合であり , まだ生成されていない t_{ij}^k に対応する .

このとき , $c_{ij}^s + u_{ij}^s + w_{ij}^{ks}$ が最小となる区分 s に関する区分フロー変数 ζ_{ij}^{ks} の被約費用は 0 となり , それ以外の品種の区分フロー変数の被約費用は非負となる . また , \bar{t}_{ij}^k は非負となる .

次に , 生成されていない双対変数 \bar{t} を含めた $PPFELR_l(\bar{P})$ の双対問題 $DU_l(\bar{P})$ を示す .

$(DU_l(\bar{P}))$

$$\text{最大化} \quad \sum_{k \in K} d^k \pi^k - \sum_{(i,j) \in A} \left(r_{ij} + \sum_{s \in S_{ij}} g_{ij}^s \right) \quad (6.164)$$

$$\text{条件} \quad \sum_{(i,j) \in A_{\bar{P}^k}} \delta_{ij} t_{ij}^k + \sum_{(i,j) \in A \setminus A_{\bar{P}^k}} \delta_{ij} \bar{t}_{ij}^k \geq \pi^k \quad \forall p \in P^k, k \in K \quad (6.165)$$

$$t_{ij}^k \leq c_{ij}^s + u_{ij}^s + w_{ij}^{ks} \quad \forall s \in S_{ij}, (i,j) \in A_{\bar{P}^k}, k \in K \quad (6.166)$$

$$\bar{t}_{ij}^k \leq c_{ij}^s + u_{ij}^s + w_{ij}^{ks} \quad \forall s \in S_{ij}, (i,j) \in A \setminus A_{\bar{P}^k}, k \in K \quad (6.167)$$

$$b_{ij}^{sl} u_{ij}^s + \sum_{k \in K} d^k w_{ij}^{ks} \leq f_{ij}^s + \frac{b_{ij}^{sl}}{b_{ij}^s} \left(r_{ij} + g_{ij}^s \right) \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.168)$$

$$u_{ij}^s \geq 0 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.169)$$

$$w_{ij}^{ks} \geq 0 \quad \forall k \in K, s \in S_{ij}, (i,j) \in A \quad (6.170)$$

$$r_{ij} \geq 0 \quad \forall (i,j) \in A \quad (6.171)$$

$$g_{ij}^s \geq 0 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.172)$$

$A \setminus A_{\bar{P}^k}$ であるアークに対して t の値を \bar{t} に設定したとき, \bar{t}_{ij}^k はアーク (i,j) の区分集合の中で $c_{ij}^s + u_{ij}^s + w_{ij}^{ks}$ の最小値であることから, 双対問題 $DU_l(\bar{P})$ の (6.167) 式を満足する. (6.165) 式の第二項は非負であるので, (6.165) 式を満足する. また, $DU_l(\bar{P})$ の目的関数には \bar{t}_{ij}^k が含まれないため, \bar{t}_{ij}^k に設定しても, 目的関数値は変化しない. このため, (6.163) 式にしたがって, \bar{t}_{ij}^k とした解は, $DU_l(\bar{P})$ の実行可能解であり, 目的関数値も同一である. したがって, 設定した \bar{t}_{ij}^k は, $DU_l(\bar{P})$ の最適解となる. 以上のことから, 生成されていない (6.162) に対応する最適な双対変数 t_{ij}^k の値は, (6.163) 式により設定できる.

一方, アークの長さを t または \bar{t} としたとき, z_p^k に関する被約費用は「パス p の長さ $-\pi^k$ 」となるため, $PPFELR_l(\bar{P})$ における品種 k に関する価格付け問題は, 次のような問題 PRP^k となる.

(PRP^k)

$$\text{最小化} \quad \sum_{p \in P^k} \left(\sum_{(i,j) \in A_{\bar{P}^k}} \delta_{ij}^p t_{ij}^k + \sum_{(i,j) \in A \setminus A_{\bar{P}^k}} \delta_{ij}^p \bar{t}_{ij}^k - \pi^k \right) z_p^k \quad (6.173)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad (6.174)$$

$$z_p^k \geq 0 \quad \forall p \in P^k \quad (6.175)$$

被約費用が負となるパスフロー変数を見つけるためには, 品種 $k (k \in K)$ に対して最短パスを探索し, 「最短距離 $-\pi^k$ 」が負であればよい. π^k は現在のパス集合における品種 k の始点・終点間の最短距離であり, 定数項である. このため, 各品種 k に対して, (6.173) 式の右辺の第一項と第二項の和を最小化するパス p を見つければよい.

この問題は, 次のようなアーク (i,j) の長さを t_{ij}^k または \bar{t}_{ij}^k とした品種 k に対する始点・終点間の最短路問題 SPP^k と等価となる.

(SPP^k)

$$\text{最小化} \quad \sum_{(i,j) \in A_{\bar{P}^k}} t_{ij}^k \nu_{ij}^k + \sum_{(i,j) \in A \setminus A_{\bar{P}^k}} \bar{t}_{ij}^k \nu_{ij}^k \quad (6.176)$$

$$\text{条件} \quad \sum_{i \in N_n^+} \nu_{in}^k - \sum_{j \in N_n^-} \nu_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N \quad (6.177)$$

$$\nu_{ij}^k \geq 0, \nu_{ji}^k \geq 0 \quad \forall (i, j) \in A \quad (6.178)$$

SPP^k を解き、品種 k の始点・終点間の最短距離 μ^k を求める。 $\mu^k - \pi^k < 0$ であれば、この最短路に対応するパスフロー変数が生成すべき z_p^k となり、このパスを \bar{P}^k に加え、 $A_{\bar{P}^k}$ を更新する。 $A_{\bar{P}^k}$ において、新たに要素となったアークの集合を $\Delta_{\bar{P}^k}$ とする。続いて、生成された z_p^k に関するまだ生成されていない品種区分フロー変数 ζ_{ij}^{ks} を生成する。 $\Delta_{\bar{P}^k}$ に対応する (6.144) 式と (6.146) 式を追加する。

PPFEL に対する列生成法と行生成法を用いた解法の流れは、次のようになる。

列生成法および行生成法

[ステップ 1] 品種 $k (\in K)$ 毎に適当な初期パス集合を求め、 \bar{P}^k とする。初期パス集合 $\bar{P}^k (k \in K)$ に対する $A_{\bar{P}^k}$ を求める。

[ステップ 2] PPFEL _{l} (\bar{P}) を解き、 π, t, u および w を求める。

[ステップ 3] すべての品種 $k (\in K)$ に対して、以下の操作を行う。

- 1) $A \setminus A_{\bar{P}^k}$ に含まれるアーク (i, j) について、 \bar{t}_{ij}^k を求める。
- 2) アーク $(i, j) (\in A)$ の長さを t^k または \bar{t}^k とした最短路問題 SPP^k を解き、品種 k の始点・終点間の最短距離 μ^k と最短路 p^k を求める。
- 3) $\mu^k - \pi^k < 0$ であれば、パスフロー変数 $z_{p^k}^k$ を生成し、 p^k を \bar{P}^k に加える。
- 4) $\Delta_{\bar{P}^k}$ を求め、 $A_{\bar{P}^k}$ を更新する。
- 5) $\zeta_{ij}^{ks} (s \in S_{ij}, (i, j) \in \Delta_{\bar{P}^k})$ を生成する。
- 6) $\Delta_{\bar{P}^k}$ について、(6.144) 式と (6.146) 式を生成し、PPFEL _{l} (\bar{P}) に加える。

[ステップ 4] 追加されたパスがあればステップ 2 へ戻る。そうでなければ終了する。

容量スケールリング法では、容量を変更して繰り返し線形緩和問題を解く。このため、二回目以降では、前回までに生成したパス集合を初期集合として利用できる。

6.4.4 近似解法

容量 b^l を変更して、PPFEL _{l} を繰り返し解き直す。得られた線形緩和問題において、多重アークの中で複数のダミーアークのフロー量が正となる場合がある。このため、単純に線形緩和問題 \tilde{y}_{ij}^s が正であるダミーアークを 1 すると、PPFE' の制約式である (6.51) 式を満足しない場合がある。そこで、線形緩和問題における総フローを $\tilde{X}_{ij} (= \sum_{s \in S_{ij}} \tilde{c}_{ij}^{ks})$ とし、 \tilde{X}_{ij} がどの区分の範囲にあるかによって、 y_{ij}^s を設定する。 \tilde{X}_{ij} が上限 b_{ij}^s と下限 b_{ij}^{s-1}

の範囲にある区分に対応する区分変数を 1, それ以外を 0 とした区分変数を \bar{y}_{ij}^s とする.

$$\bar{y}_{ij}^s = \begin{cases} 1 & \text{if } b_{ij}^{s-1} < \tilde{X}_{ij} \leq b_{ij}^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_{ij}, (i, j) \in A \quad (6.179)$$

このとき, \bar{y} は (6.51) 式と (6.53) 式を満足する

次に, 区分変数を \bar{y} としたネットワーク上で, 次のような上下限付きの多品種フロー問題 $MCFLU(\bar{y})$ を作成する.

($MCFLU(\bar{y})$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \xi_{ij}^s + f_{ij}^s \bar{y}_{ij}^s) \quad (6.180)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (6.181)$$

$$X_{ij} = \sum_{k \in K} x_{ij}^k \quad \forall (i, j) \in A \quad (6.182)$$

$$X_{ij} = \sum_{s \in S_{ij}} \xi_{ij}^s \quad \forall (i, j) \in A \quad (6.183)$$

$$\begin{cases} b_{ij}^{s-1} < \xi_{ij}^s \leq b_{ij}^s & \text{if } \bar{y}_{ij}^s = 1 \\ \xi_{ij}^s = 0 & \text{if } \bar{y}_{ij}^s = 0 \end{cases} \quad \forall s \in S_{ij}, (i, j) \in A \quad (6.184)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6.185)$$

$MCFLU(\bar{y})$ はアークフロー変数による定式化ではあるが, 多品種フロー問題であるため, 汎用の数理最適化ソフトウェアを用いて解くことができる. アークフローによる定式化を用いるのは, パスフローの定式化では, すべてのパスが生成されていないためである. $MCFLU(\bar{y})$ の実行可能解 $\tilde{\xi}$ を求めることができれば, \bar{y} および $\tilde{\xi}$ は PND の実行可能解となる.

汎用の数理最適化ソフトウェアの分枝限定法を用いると, 少数の 0-1 変数をもつ計画問題に対しては比較的短時間で最適解を求めることができるが, $PPFE'$ は $|A||S|$ 個の 0-1 変数をもつため, 直接解くことは困難である. 一方, 容量スケールリング法では, 繰り返し回数とともに, 多くの 0-1 変数である区分変数が 0 または 1 に収束する傾向がある. そこで, $PPFEL_l$ の区分変数解を \tilde{y} とし, 繰り返し回数 l において, 0 または 1 に収束していない区分変数の数 ω_l が決められた数 ψ 以下となれば, 次のような区分変数の範囲を限定した条件を付加した問題 $PPFER_l(\bar{P}, \tilde{y})$ を作成する.

$$\sum_{s \in S_{ij}} y_{ij}^s = 0 \quad \text{if } \sum_{s \in S_{ij}} \tilde{y}_{ij}^s < \epsilon_0 \quad (i, j) \in A \quad (6.186)$$

ここで, ϵ_0 は $0 < \epsilon_0 < 1$ である定数である.

$PPFER_l(\bar{P}, \tilde{y})$ は, 次のようになる.

$(PPFER_l(\bar{P}, \tilde{y}))$

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{s \in S_{ij}} (c_{ij}^s \sum_{k \in K} \zeta_{ij}^{ks} + f_{ij}^s y_{ij}^s) \quad (6.187)$$

$$\text{条件} \quad \sum_{p \in \bar{P}^k} z_p^k = d^k \quad \forall k \in K \quad (6.188)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k = \sum_{s \in S_{ij}} \zeta_{ij}^{ks} \quad \forall (i,j) \in A_{\bar{P}^k}, k \in K \quad (6.189)$$

$$0 \leq \sum_{k \in K} \zeta_{ij}^{ks} \leq b_{ij}^{sl} y_{ij}^s \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.190)$$

$$0 \leq \zeta_{ij}^{ks} \leq d^k y_{ij}^s \quad \forall s \in S_{ij}, (i,j) \in A_{\bar{P}^k}, k \in K \quad (6.191)$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \quad \forall (i,j) \in A \quad (6.192)$$

$$\begin{cases} \sum_{s \in S_{ij}} y_{ij}^s = 0 & \text{if } \sum_{s \in S_{ij}} \tilde{y}_{ij}^s \leq \epsilon_0 \\ \sum_{s \in S_{ij}} y_{ij}^s \leq 1 & \text{otherwise} \end{cases} \quad (i,j) \in A \quad (6.193)$$

$$0 \leq y_{ij}^s \leq 1 \quad \forall s \in S_{ij}, (i,j) \in A \quad (6.194)$$

$$z_p^k \geq 0 \quad \forall p \in \bar{P}^k, k \in K \quad (6.195)$$

$PPFER_l(\bar{P}, \tilde{y})$ に、分枝限定法を適用すれば、 PND の近似解を求めることができる。しかし、 $PPFER_l(\bar{P}, \tilde{y})$ のように 0-1 変数を限定した問題であっても、分枝限定法によって最適解を求めることが困難な場合がある。そのため、実際の計算では、分枝限定法は一定時間で打ち切り、その時点で得られた最良の上界値を採用する。また、 ψ を一定値とした場合、連続した繰り返し回数において分枝限定法を繰り返し同一の解を生成することが多くなるため、これを防ぐために分枝限定法を適用する毎に ψ を β だけ減少させ、解の多様性を確保する。

次に $PPFE$ に対する近似解法を含めた容量スケールリング法を示す。

容量スケールリング法

[ステップ 1] スケールリングパラメータを λ 、繰り返し回数の上限を l_{max} 、繰り返し回数の下限を l_{min} 、閾値を ϵ_0, ϵ_1 、および ϵ_2 、最良の上界値を UB 、分枝限定法の適用基準を ψ 、その変更量を β とする。 $b^l := b$ 、 $UB := \infty$ 、 $l := 0$ とする。

[ステップ 2] $l := l + 1$ とする。列生成法と行生成法を用いて $PPFEL_l$ を解き、最適解 \tilde{y} を求める。

[ステップ 3] $b^l := \lambda b^{l-1} \tilde{y} + (1 - \lambda) b^{l-1}$ または $b^l := \{1 - \lambda(1 - \tilde{y})\} b^{l-1}$ とする。

[ステップ 4] \tilde{y} を求め、 $MCFLU(\tilde{y})$ を解き、上界値 UB_l を求める。 $UB_l \leq UB$ であれば、 $UB := UB_l$ とする。

[ステップ 5] $\epsilon_1 < \tilde{y}_{ij}^s < \epsilon_2 ((i,j) \in A)$ である区分変数の数 ω_l を求める。 $\omega_l \leq \psi$ であれば、 ϵ_0 を用いて $PPFER_l(\bar{P}, \tilde{y})$ を作成し、分枝限定法を適用し、上界値 UB_R を求める。 $UB_R \leq UB$ であれば、 $UB := UB_R$ とし、 $\psi := \psi - \beta$ とする。

[ステップ 6] $l = l_{max}$ または $l > l_{min}$ かつ $\omega_l \leq \psi$ であれば終了, そうでなければステップ 2 へ戻る.

6.5 数値実験

提案した容量スケールリング法を評価するために数値実験を行う. Crainic らのベンチマーク問題 [36, 37] をもとに区分的線形費用をもつネットワーク設計問題に対する問題を作成し, Croxton-Gendron-Magnanti が [24] が示した従来の解法と容量スケールリング法によって求めた上界値を比較する. Crainic らのベンチマーク問題は C 問題と R 問題に分かれている. 本章では, これらの問題の内, 規模の小さな C 問題 27 問, R 問題 126 問を対象として数値実験を行う. なお, 使用したデータは www.rku.ac.jp/~katayama/networkdata に掲載している.

Croxton らの解法は, アークフローによる定式化と同様の三つの定式化 *PAFB*, *PAFS*, *PAFE* に対する閾値と分枝限定法を組み合わせた近似解法 (以下, 閾値法) である. 閾値法は, *PAFB*, *PAFS* および *PAFE* の線形緩和問題を解き, 求めた区分変数の線形緩和解が 0 であれば区分変数を 0 に固定し, 閾値 (0.7) 以上であれば区分変数を 1 に固定し, それ以外の区分変数を 0-1 の自由変数に設定した問題に対して, 分枝限定法を適用する方法である.

主な条件およびパラメータは次の通りである.

- 1) 問題数: C 問題 27 問, R 問題 126 問
- 2) ノード数: C 問題 20~100, R 問題 10~20
- 3) アーク数: C 問題 100~400, R 問題 35~220
- 4) 品種数: C 問題 10~200, R 問題 10~200
- 5) 費用: C 問題 $F \cdot V$, R 問題 $c_1 \cdot c_2 \cdot c_8$
- 6) 区分: C 問題 $T \cdot L$, R 問題 $f_{01} \cdot f_{05} \cdot f_{10}$
- 7) データ: Crainic らの C 問題および R 問題をもとに, 次のようにデータを作成する.

- 区分数: すべてのアークに対して, 3 とする.
- 区分の上限值:

$$b_{ij}^s = \frac{(2^s - 1)b_{ij}}{7}, s = 1, 2, 3, (i, j) \in A$$

b_{ij} は元のデータのアーク容量とする.

- 変動費用: $c_{ij} \times (0.7)^{s-1}$ とする. c_{ij} は元のデータの変動費用とする.
- 固定費用: 区分間において連続となるように設定する. 第 1 区分の固定費用は元のデータの固定費用とする.

- 8) 容量の変更方法: $C^l := \{1 - \lambda(1 - \tilde{y})\}C^{l-1}$
- 9) アークフローによる定式化: パラメータ

- スケーリングパラメータ λ : 0.25 とする .
 - 閾値 $\epsilon_0, \epsilon_1, \epsilon_2$: すべて 0.01 とする .
 - 分枝限定法の適用基準の初期値 ψ : 75 とする .
 - 分枝限定法の適用基準の変更量 β : 10 とする .
 - 計算時間 : 計算時間の上限を 10 時間とし , 上限を超える場合は分枝限定法の終了時点で計算を打ち切り , その時点の最良解を採用する . 分枝限定法の 1 回当たりの計算時間の上限を 1 時間とする . なお , 従来の解法である閾値法では , 計算時間が 10 時間を超える場合は分枝限定法の終了時点で計算を打ち切る .
 - 最大繰り返し回数 l_{max} : 200 回とする . ただし , 計算時間が 10 時間を超える場合は , 計算を打ち切る .
- 10) パスフローによる定式化 : パラメータ
- スケーリングパラメータ λ : 0.05 ~ 0.25 とする .
 - 閾値 $\epsilon_0, \epsilon_1, \epsilon_2$: すべて 0.01 とする .
 - 分枝限定法の適用基準の初期値 ψ : 75 とする .
 - 分枝限定法の適用基準の変更量 β : 10 とする .
 - 分枝限定法の 1 回当たりの計算時間の上限を 30 分とする .
 - 最小繰り返し回数 l_{min} : 50 回とする .
 - 最大繰り返し回数 l_{max} : 200 回とする .
- 11) 誤差 : $(\text{上界値} - \text{下界値}) / \text{下界値} \times 100(\%)$
である . 下界値は CPLEX により求める . CPLEX の計算時間の上限は 10 時間とし , 上限を超える場合はその時点の最良値を採用する .
- 12) 閾値法の計算時間 : 計算時間の上限は 10 時間とし , 上限を超える場合はその時点の最良値を採用する .
- 13) 言語 : Visual Studio 2005 C++.NET
- 14) 数理最適化ソフトウェア : CPLEX 11.2
- 15) 使用コンピュータ : CPU Pentium Core2-Duo 3.3GHz ,メモリ 4GB ,OS Windows XP

表 6.1 C 問題の属性

問題	ノード数	アーク数	品数	費用	区分	問題	ノード数	アーク数	品数	費用	区分
C1	25	100	10	V	L	C14	20	230	200	V	L
C2	25	100	10	F	L	C15	20	230	200	F	L
C3	25	100	10	F	T	C16	20	230	200	V	T
C4	25	100	30	V	T	C17	20	230	200	F	T
C5	25	100	30	F	L	C18	20	300	200	V	L
C6	25	100	30	F	T	C19	20	300	200	F	L
C7	20	230	40	V	L	C20	20	300	200	V	T
C8	20	230	40	V	T	C21	20	300	200	F	T
C9	20	230	40	F	T	C22	100	400	10	V	L
C10	20	300	40	V	L	C23	100	400	10	F	L
C11	20	300	40	F	L	C24	100	400	10	F	T
C12	20	300	40	V	T	C25	100	400	30	V	T
C13	20	300	40	F	T	C26	100	400	30	F	L
						C27	100	400	30	F	T

表 6.2 C 問題:実行可能解数

解法		実行可能解数 (全 27 問)
閾値	<i>PAFB</i>	14
	<i>PAFS</i>	17
	<i>PAFE</i>	21
	全体	23
容量スケーリング	<i>PAFB</i>	27
	<i>PAFS</i>	27
	<i>PAFE</i>	27
	<i>PPFE</i>	27

表 6.3 C 問題:解法別平均誤差 *¹

解法		誤差 (%)
閾値法	最良値	5.33
アークフロー	<i>PAFB</i>	5.43
	<i>PAFS</i>	4.09
	<i>PAFE</i>	3.39
	最良値	3.22
パスフロー: <i>PPFE</i>	0.05	5.27
	0.10	4.09
	0.15	3.97
	0.20	3.89
	0.25	4.72
	最良値	3.52

*1:閾値法で実行可能解が求められた 23 問

表 6.4 C 問題:性質別平均誤差 *¹

費用	区分	閾値	アークフロー			パスフロー	
		最良値 (%)	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	最良値 (%)	<i>PPFE</i> (%)
V	T	3.76	3.27	2.30	2.09	2.02	1.98
V	L	3.20	4.12	2.55	2.57	2.31	2.36
F	T	4.25	5.74	4.77	3.96	3.95	4.26
F	L	9.74	8.36	6.49	4.81	4.46	5.30
平均		5.33	5.43	4.09	3.39	3.22	3.52

*1:閾値法で実行可能解が求められた 23 問

表 6.5 C 問題:規模別平均誤差 *1

N , A , K	閾値	アークフロー			バスフロー	
	最良値 (%)	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	最良値 (%)	<i>PPFE</i> (%)
25,100,10	13.98	2.50	1.07	0.00	0.00	0.00
25,100,30	1.69	0.45	0.52	0.20	0.15	0.20
20,230,40	1.49	0.89	0.97	0.22	0.22	0.39
20,300,40	1.21	2.16	0.46	0.10	0.10	0.39
20,230,200	5.56	8.68	5.76	5.95	5.76	5.68
20,300,200	4.30	7.83	4.71	5.43	4.70	5.01
100,400,30	13.60	13.70	14.61	10.26	10.26	11.62
平均	5.33	5.43	4.09	3.39	3.22	3.52

*1:閾値法で実行可能解が求められた 23 問

表 6.6 C 問題:上界値の比較

問題	最適値/ 下界値	閾値			アークフロー			パスフロー
		<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	<i>PPFE</i>
C1	13013 ^O	—	—	13497	13160	13160	13013	13013
C2	14358 ^O	—	17839	—	14914	14504	14358	14358
C3	47991 ^O	—	—	—	48608	48608	48608	49206
C4	306100 ^O	—	308554	308501	306228	306100	306100	306100
C5	36097 ^O	—	37489	37856	36258	36310	36310	36310
C6	81125 ^O	—	—	81474	81818	81917	81125	81141
C7	351996 ^O	482974	—	354065	357435	351996	351996	352473
C8	273835 ^O	—	286218	278391	273835	277702	273835	274264
C9	489912 ^L	519085	525241	500819	495449	497284	493103	494267
C10	360403 ^O	412829	—	365240	362398	362118	361841	361461
C11	472638 ^O	558531	—	481947	499920	478354	472638	478027
C12	350315 ^O	366044	—	354883	350832	350352	350315	350549
C13	470226 ^O	499377	479950	471202	480357	470817	470277	470533
C14	77372 ^L	82932	82339	82386	85434	82356	82500	82629
C15	116897 ^L	125813	125205	125689	127942	125133	125610	125023
C16	80085 ^L	84093	83412	83673	85944	83928	84021	83412
C17	115241 ^L	121938	120482	120473	123936	120741	120786	120786
C18	62832 ^L	66305	65294	65407	67202	65791	66485	65703
C19	98223 ^L	104706	104385	104187	109028	104344	105844	105740
C20	63755 ^L	66422	65792	65824	68686	65930	66122	66006
C21	90956 ^L	94999	94878	94592	96089	95041	94994	94869
C22	23652 ^L	—	—	—	24017	24016	23863	23863
C23	22513 ^L	—	—	—	27540	24052	23357	23357
C24	55044 ^L	—	—	—	65513	70645	69176	71076
C25	310445 ^L	—	346045	—	323968	323410	322528	322737
C26	43966 ^L	—	55965	50652	52595	54023	49705	50759
C27	117940 ^L	—	134582	134886	138140	137720	134265	136172

O:最適値; L:下界; —:実行可能解なし

表 6.7 C 問題:上界値の誤差の比較

問題	閾値			アークフロー				パスフロー	
	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	改善率 (%)	<i>PPFE</i> (%)	改善率 (%)
C1	—	—	3.72	1.13	1.13	0.00	-3.72	0.00	-3.72
C2	—	24.24	—	3.87	1.02	0.00	-24.24	0.00	-24.24
C3	—	—	—	1.29	1.29	1.29	—	2.53	—
C4	—	0.80	0.78	0.04	0.00	0.00	-0.78	0.00	-0.78
C5	—	3.86	4.87	0.45	0.59	0.59	-3.41	0.59	-3.27
C6	—	—	0.43	0.85	0.98	0.00	-0.43	0.02	-0.41
C7	37.21	—	0.59	1.55	0.00	0.00	-0.59	0.14	-0.45
C8	—	4.52	1.66	0.00	1.41	0.00	-1.66	0.16	-1.51
C9	5.95	7.21	2.23	1.13	1.50	0.65	-1.57	0.89	-1.34
C10	14.55	—	1.34	0.55	0.48	0.40	-0.94	0.29	-1.05
C11	18.17	—	1.97	5.77	1.21	0.00	-1.97	1.14	-0.83
C12	4.49	—	1.30	0.15	0.01	0.00	-1.30	0.07	-1.24
C13	6.20	2.07	0.21	2.15	0.13	0.01	-0.20	0.07	-0.14
C14	7.19	6.42	6.48	10.42	6.44	6.63	0.02	6.79	0.37
C15	7.63	7.11	7.52	9.45	7.05	7.45	-0.06	6.95	-0.16
C16	5.01	4.15	4.48	7.32	4.80	4.92	0.64	4.15	0.00
C17	5.81	4.55	4.54	7.55	4.77	4.81	0.23	4.81	0.27
C18	5.53	3.92	4.10	6.95	4.71	5.81	0.79	4.57	0.65
C19	6.60	6.27	6.07	11.00	6.23	7.76	0.16	7.65	1.58
C20	4.18	3.19	3.24	7.73	3.41	3.71	0.22	3.53	0.34
C21	4.44	4.31	4.00	5.64	4.49	4.44	0.44	4.30	0.30
C22	—	—	—	1.54	1.54	0.89	—	0.89	—
C23	—	—	—	22.33	6.84	3.75	—	3.75	—
C24	—	—	—	19.02	28.34	25.68	—	29.13	—
C25	—	11.47	—	4.36	4.18	3.89	-7.58	3.96	-7.51
C26	—	27.29	15.21	19.63	22.88	13.05	-2.15	15.45	0.24
C27	—	14.11	14.37	17.13	16.77	13.84	-0.27	15.46	1.35

—:実行可能解なし,または算出できず

表 6.8 C 問題:区分スケーリングの繰返し回数

問題	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	<i>PPFE</i>
C1	82	42	39	39
C2	200	110	76	52
C3	113	81	112	68
C4	47	125	109	50
C5	179	53	46	52
C6	78	116	137	64
C7	170	162	126	84
C8	130	86	110	51
C9	200	117	175	79
C10	130	87	84	142
C11	200	84	122	55
C12	189	142	108	71
C13	200	137	89	55
C14	200	50	18	54
C15	129	184	16	90
C16	162	94	27	67
C17	149	200	19	83
C18	200	136	17	94
C19	200	157	10	144
C20	97	96	27	68
C21	121	84	15	102
C22	185	109	93	149
C23	200	117	136	74
C24	200	188	200	118
C25	104	96	67	57
C26	200	132	83	63
C27	200	200	127	127

表 6.9 C 問題:性質別計算時間 *¹

費用	区分	閾値	アークフロー			パスフロー
		最大値 (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
V	T	12096.0	7449.0	13044.5	18370.5	1474.6
V	L	14404.2	6827.9	9680.3	14516.3	3025.3
F	T	18011.4	9770.3	11159.4	19001.8	3205.7
F	L	18016.3	7587.4	10700.8	17439.0	7097.1
平均		15685.4	7955.6	11210.0	17454.3	3730.0

*1:閾値法で実行可能解が求められた 23 問

表 6.10 C 問題:規模別計算時間 *1

N , A , K	閾値	アークフロー			パスフロー
	最大値 (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
25,100,10	2.0	1.6	8.1	21.4	12.9
25,100,30	15.5	95.7	481.7	3936.9	721.4
20,230,40	19.6	53.0	178.7	1063.1	89.7
20,300,40	24.7	113.4	253.9	1004.9	144.9
20,230,200	36000.0	19268.2	24444.5	36000.0	9131.3
20,300,200	36000.0	19794.8	26166.5	36000.0	10019.3
100,400,30	24185.0	8608.4	17457.6	31462.3	2050.1
平均	15685.4	7955.6	11210.0	17454.3	3730.0

*1:閾値法で実行可能解が求められた 23 問

表 6.11 C 問題:計算時間

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
C1	—	—	0.3	0.8	1.0	1.9	1.1
C2	—	3.6	—	2.4	15.3	41.0	24.7
C3	—	—	—	5.1	14.1	180.5	58.7
C4	—	1.9	2.6	93.3	737.1	636.5	29.6
C5	—	8.7	31.9	58.9	417.5	8164.9	1854.4
C6	—	—	12.0	134.9	290.7	3009.3	280.1
C7	2.7	—	3.2	26.4	132.3	126.0	13.0
C8	—	6.7	2.9	29.6	150.0	332.3	11.9
C9	6.0	8.6	48.9	103.0	253.8	2731.1	244.1
C10	4.3	—	17.5	70.1	114.3	453.4	26.9
C11	2.9	—	62.3	125.2	202.5	2041.2	518.2
C12	3.1	—	11.4	172.6	508.3	1254.1	24.7
C13	7.7	3.0	7.7	85.5	190.5	270.7	9.9
C14	<i>t</i>	<i>t</i>	<i>t</i>	22495.0	20663.9	<i>t</i>	7721.2
C15	<i>t</i>	<i>t</i>	<i>t</i>	18886.0	27506.5	<i>t</i>	14510.0
C16	<i>t</i>	<i>t</i>	<i>t</i>	9396.0	21262.9	<i>t</i>	4367.7
C17	<i>t</i>	16270.5	14033.2	26295.8	28344.7	<i>t</i>	9926.3
C18	<i>t</i>	<i>t</i>	<i>t</i>	11547.0	27490.2	<i>t</i>	7364.2
C19	<i>t</i>	<i>t</i>	<i>t</i>	25942.5	32578.2	<i>t</i>	21680.8
C20	<i>t</i>	<i>t</i>	<i>t</i>	12920.7	19608.8	<i>t</i>	3431.3
C21	<i>t</i>	6701.9	16876.0	28769.1	24988.7	<i>t</i>	7600.8
C22	—	—	—	13.4	28.4	14.0	5.1
C23	—	—	—	39.3	200.5	546.6	424.6
C24	—	—	—	42.7	155.4	663.2	72.7
C25	—	555.1	—	22081.6	<i>t</i>	<i>t</i>	982.6
C26	—	<i>t</i>	<i>t</i>	509.7	3484.6	22386.8	3994.7
C27	—	<i>t</i>	<i>t</i>	3233.8	12888.1	<i>t</i>	1172.9

t:計算時間の上限; —:実行可能解なし

表 6.12 R 問題:属性 1

問題	ノード	アーク	品種	問題	ノード	アーク	品種
R01	10	35	10	R10	20	120	40
R02	10	35	25	R11	20	120	100
R03	10	35	50	R12	20	120	200
R04	10	60	10	R13	20	220	40
R05	10	60	25	R14	20	220	100
R06	10	60	50	R15	20	220	200
R07	10	83	10				
R08	10	83	25				
R09	10	83	50				

表 6.13 R 問題:属性 2

問題	区分	費用
R*-1	c1	f01
R*-2	c1	f05
R*-3	c1	f10
R*-4	c2	f01
R*-5	c2	f05
R*-6	c2	f10
R*-7	c8	f01
R*-8	c8	f05
R*-9	c8	f10

Crainic らのベンチマーク問題をもとに作成した問題を用いて、三つの定式化 $PAFB$, $PAFS$, $PAFE$ に対する閾値法と四つの定式化 $PAFB$, $PAFS$, $PAFE$, $PPFE$ に対する容量スケール法を比較する。はじめに C 問題の計算結果を示し、続いて R 問題の計算結果を示す。

C 問題の属性を表 6.1 に示す。表の数値は、ノード数、アーク数、品種数、および費用と区分の性質である。C 問題は、費用および区分の上限値がそれぞれ二種類に分類され、F は固定費用が変動費用に比べて相対的に高い、V は固定費用が相対的に安い問題であり、T は区分の上限値が小さい、L は区分の上限値が大きい問題である。

表 6.2 に、C 問題に対して実行可能解が求められた問題数を示す。ここでは、アークフローによる定式化 $PAFB$ を用いた解法を $PAFB$ 、 $PAFS$ を用いた解法を $PAFS$ 、 $PAFE$ を用いた解法を $PAFE$ とし、パスフローによる定式化 $PPFE$ を用いた解法を $PPFE$ で表す。閾値法では、 $PAFB$ は 27 問中 14 問、 $PAFS$ は 17 問、 $PAFE$ で 21

表 6.14 R 問題:性質別平均誤差 *1

区分	費用	閾値	アークフロー			パスフロー	
		最良値 (%)	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	最良値 (%)	<i>PPFE</i> (%)
c1	f01	0.67	1.26	0.43	0.34	0.32	0.41
c1	f05	2.66	3.92	1.85	1.65	1.47	1.64
c1	f10	3.05	6.08	2.18	2.43	2.11	2.55
c2	f01	1.25	1.79	0.49	0.27	0.27	0.34
c2	f05	2.69	4.95	1.30	0.87	0.83	1.11
c2	f10	4.10	5.68	1.90	1.68	1.56	1.80
c8	f01	4.33	1.75	0.84	0.49	0.49	0.48
c8	f05	1.68	2.94	1.83	0.95	0.87	0.93
c8	f10	2.57	2.83	1.78	1.49	1.34	1.72
平均		2.54	3.54	1.39	1.13	1.03	1.23

*1:閾値法で実行可能解が求められた 123 問

問において実行可能解が求められた。しかし、*PAFB* は 27 問中 13 問、*PAFS* は 10 問、*PAFE* は 6 問において実行可能解が求められず、三つの定式化ともに実行可能解が求められない問題が 4 問 (C3, C22, C23, C24) 存在した。実行可能解が求められない場合があることは、Croxtton-Gendron-Magnanti[24] も指摘している。一方、容量スケールリング法では、すべての問題に対して実行可能解を求めることができた。

表 6.3 は、C 問題に対して各解法により求められた上界値の誤差の平均値である。閾値法では実行可能解が求められない場合があるため、三つの定式化によるものの最良値を示す。また、4 問は閾値法のどの定式化でも上界値が求められないために、集計の対象から除外している。閾値法の平均誤差は 5.33% である。*PAFB* の平均誤差は 5.43%、*PAFS* の平均誤差は 4.09%、*PAFE* の平均誤差は 3.39%、これらの最良値の平均誤差は 3.22% であり、閾値法との差は 2.11% である。また、*PPFE* の平均誤差はスケールリングパラメータにより 3.89~5.27%、これらの最良値の平均誤差は 3.52% であり、閾値法との差は 1.81% である。容量スケールリング法のいずれの解法も、閾値法と同等または良い解を算出しており、実行可能解が算出できなかった 4 問を含めると、さらに差があるものとする。

表 6.4 は、C 問題に対する費用と区分の性質別の平均誤差である。閾値法の誤差は、固定費用が高く区分の大きな FL 問題に対して 9.74% と大きく、それ以外の問題に対して 3.20~4.25% である。容量スケールリング法の *PAFB* の誤差は、FL 問題に対して 8.36% と大きく、それ以外の問題に対して 3.27~5.74% である。*PAFS* の誤差は、FL 問題に対して 6.49% と大きく、それ以外の問題に対して 2.30~4.77% である。*PAFE* の誤差は FL 問題に対しては比較的小さく、性質別の誤差は 2.09~4.81% である。アークフローによ

表 6.15 R 問題:規模別平均誤差

N , A , K	閾値	アークフロー			パスフロー	
	最良値 (%)	<i>PAFB</i> (%)	<i>PAFS</i> (%)	<i>PAFE</i> (%)	最良値 (%)	<i>PPFE</i> (%)
10,35,10	2.04	0.48	0.22	0.00	0.00	0.00
10,35,25	1.51	0.00	0.00	0.00	0.00	0.00
10,35,50	0.13	0.00	0.00	0.00	0.00	0.00
10,60,10	2.67	3.83	0.54	0.60	0.48	0.61
10,60,25	4.43	1.65	0.08	0.18	0.00	0.25
10,60,50	2.87	1.33	0.37	0.19	0.19	0.37
10,83,10	4.22	2.79	1.36	0.62	0.62	0.84
10,83,25	2.12	2.78	1.11	0.59	0.58	0.84
10,83,50	0.80	1.32	0.14	0.06	0.04	0.09
20,120,40	2.71	5.61	0.49	0.65	0.31	1.02
20,120,100	1.43	3.19	1.66	1.19	1.17	1.07
20,120,200	1.25	2.34	1.60	1.48	1.44	1.27
20,220,40	3.03	8.92	3.29	1.88	1.88	2.41
20,220,100	4.50	8.58	5.34	4.73	4.58	5.03
20,220,200	3.04	6.81	3.13	3.53	3.00	3.18
平均	2.54	3.54	1.39	1.13	1.03	1.23

*1:閾値法で実行可能解が求められた 123 問

る定式化を用いて算出した最良値の誤差は 2.02 ~ 4.46% である。また、*PPFE* の性質別の誤差は 1.98 ~ 5.30% である。*PAFE* および *PPFE* は、ほぼすべての分類において閾値法よりも良い解を算出している。

表 6.5 は、C 問題に対する規模別の平均誤差である。いずれの解法も、アーク数および品種数が増加すると誤差が大きくなる傾向があるが、ノード数 25、品種数 10 の閾値法の誤差は 13.98% と大きい。これは、閾値法では *PAFS* のみでしか C2 の上界値を得られていないことが原因である。ノード数 25 以下、品種数 40 以下の問題では、閾値法の誤差は 1.21 ~ 13.98%、容量スケールリング法の *PAFB* の誤差は 0.45 ~ 2.50%、*PAFS* の誤差は 0.46 ~ 1.07%、*PAFE* の誤差は 0.00 ~ 0.22%、最良値の誤差は 0.00 ~ 0.22% である。また、*PPFE* の誤差は 0.00 ~ 0.39% である。品種数 200 の問題に対して、閾値法の誤差は 4.30 ~ 5.56%、*PAFB* の誤差は 7.83 ~ 8.68%、*PAFS* の誤差は 4.71 ~ 5.76%、*PAFE* の誤差は 5.43 ~ 5.95%、最良値の誤差は 4.70 ~ 5.76% である。また、*PPFB* の誤差は 5.01 ~ 5.68% である。一方、アーク数 100 の問題に対しては、いずれの解法でも誤差が大きく、10.26 ~ 13.60% である。これは、CPLEX による下界値自体が悪く、下界値と最適値の差が大きいためと考えられる。品種数の少ない問題に対しては容量スケール

表 6.16 R 問題:性質別最適解数

区分	費用	問題数	閾値	アークフロー				パスフロー
			最良値	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	最良値	<i>PPFE</i>
c1	f01	15	3	6	9	10	12	9
c1	f05	15	2	4	8	8	8	8
c1	f10	15	3	3	9	8	9	9
c2	f01	15	0	5	6	11	11	7
c2	f05	15	0	3	4	7	7	6
c2	f10	15	1	2	2	5	4	4
c8	f01	12	0	3	5	5	5	5
c8	f05	12	1	4	4	4	5	4
c8	f10	12	0	5	6	2	6	1
合計		126	10	35	53	60	67	53

ングが良い解を算出しているが、品種数の多い問題に対しては閾値法が良い解を算出している。

表 6.6 と表 6.7 は、個別の C 問題に対する結果の比較である。表 6.6 の数値は、CPLEX による最適値または下界値、閾値法および容量スケールリング法による上界値である。なお、表内の O は最適値、 L は下界値であることを表し、 $-$ は実行可能解が求められていないことを表す。表 6.7 における誤差は、解法によって得られた上界値と最適値または下界値との差である。また、改善率は閾値法と容量スケールリング法による誤差の差であり、改善率がマイナスである問題は容量スケールリング法により閾値法よりも良い解が求められたことを表す。また、 $-$ は実行可能解が求められていない、または改善率が算出できないことを表す。

誤差が 0.00% の問題は、解法によって最適解が求められたことを示す。閾値法では 1 問も最適解が求められていないが、容量スケールリング法では 27 問の内の 8 問で最適解が求められている。また、実行可能解が得られた 23 問の内、容量スケールリング法は 17 問、閾値法は 7 問の最良値を算出している。閾値法で解が求められなかった問題を含めると、容量スケールリング法は 27 問の内の 21 問の最良値を算出している。

閾値法は、C14 ~ C21 の実行可能解を算出しているが、品種数の小さな C1 ~ C6 およびノード数の多い C22 ~ C25 では実行可能解が求められていない場合が多く、特に C3, C22, C23, C24 では三つの定式化ともに実行可能解が求められていない。

容量スケールリング法では、C1 ~ C14 までの誤差は小さいが、ノード数が大きく、固定費用が高い問題に対する誤差が大きい。これは、規模の大きな問題では CPLEX により求めた下界値が悪く、最適値と下界値の差が誤差に含まれているためと考えられる。アークフローによる定式化およびパスフローによる定式化ともに、C1 ~ C13 および C25 ~ C27 に対しては閾値法よりも良い解を算出している。しかし、C14 ~ C21 に対しては閾値法が

表 6.17 R 問題:規模別最適解数

N , A , K	問題数	閾値	アークフロー			パスフロー	
		最良値	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	最良値	<i>PPFE</i>
10,35,10	6	2	4	5	6	6	6
10,35,25	6	1	6	6	6	6	6
10,35,50	6	2	6	5	6	5	6
10,60,10	9	3	1	4	4	5	4
10,60,25	9	0	3	8	6	9	6
10,60,50	9	1	4	5	7	7	4
10,83,10	9	1	2	4	4	4	5
10,83,25	9	0	1	3	4	4	3
10,83,50	9	0	1	4	5	6	3
20,120,40	9	0	0	3	2	4	1
20,120,100	9	0	3	3	3	4	3
20,120,200	9	0	3	3	5	5	5
20,220,40	9	0	1	0	2	2	1
20,220,100	9	0	0	0	0	0	0
20,220,200	9	0	0	0	0	0	0
合計	126	10	35	53	60	67	53

良い解を算出している。

表 6.8は、容量スケールリング法におけるスケールリングの繰り返し回数である。問題規模の大きな C14~C21 に対して、*PAFE* における繰り返し回数は計算時間の上限から 10~27 回となり、十分な繰り返し回数を確保することができていない。一方、パスフローによる定式化である *PPFE* における繰り返し回数は、C14~C21 に対しても十分な繰り返し回数を確保できている。

表 6.9は、C 問題に対する費用と区分の性質別の計算時間である。解法に共通した費用と区分による顕著な差は見られない。閾値法は、いずれの問題でも計算時間が大きく、12096.0~18016.3 秒で平均 15685.4 秒、約 4.4 時間を要している。*PAFB* の性質別の計算時間は 6827.9~9770.3 秒で平均 7955.6 秒、*PAFS* の計算時間は 9680.3~13044.5 秒で平均 11210.0 秒、*PAFE* の計算時間は 14516.3~18370.5 秒で平均 17454.3 秒となり、定式化が複雑になるにしたがって計算時間が増加している。一方、*PPFE* の計算時間は、1474.6~7097.1 秒で平均 3730.3 秒、約 1 時間である。*PPFE* の計算時間は、閾値法の 1/4.2、*PAFE* の 1/4.7 となり、列生成法により大幅に計算時間を短縮することができている。また、閾値法や *PAFE* では多くの問題で計算時間の上限である 10 時間に達しているため、計算を打ち切っており、実際の計算時間の差はさらに大きなものとなる。

表 6.10は、C 問題に対する規模別の計算時間である。ノード数 25 以下、品種数 40 以

表 6.18 R 問題:性質別平均計算時間 *¹

区分	費用	閾値	アークフロー			パスフロー
		最大値	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	<i>PPFE</i>
c1	f01	2610.5	2624.3	2517.2	5648.7	660.8
c1	f05	5089.6	2061.0	2427.4	7230.5	1526.6
c1	f10	5386.8	3395.0	3370.0	7951.3	1915.4
c2	f01	2511.9	2580.3	2487.4	5963.8	466.7
c2	f05	3160.4	3433.1	2543.7	6189.5	1044.3
c2	f10	3168.4	3056.6	2639.1	7971.5	2311.2
c8	f01	183.6	1931.5	3738.7	10403.3	1942.3
c8	f05	471.7	1305.2	1099.8	7403.2	1277.6
c8	f10	698.0	3298.3	975.9	8888.0	1625.6
平均		2739.3	2664.1	2456.7	7418.0	1404.9

*1:閾値法で実行可能解が求められた 123 問

下の問題に対する計算時間は小さく、品種数 200 の問題に対する計算時間は極端に大きい。閾値法の計算時間は、ノード数 25 以下、品種数 40 以下の小さな問題に対して 2.0 ~ 24.7 秒であるが、それ以外の問題に対して 24185.0 ~ 36000 秒となり、品種数 200 の問題に対してすべての問題で計算時間の上限に達している。*PAFB* の計算時間は、小さな問題に対して 1.6 ~ 113.4 秒、それ以外の問題に対して 608.4 ~ 19794.8 秒である。*PAFS* の計算時間は、小さな問題に対して 8.1 ~ 253.9 秒、それ以外の問題に対して 17457.6 ~ 26166.5 秒である。*PAFE* の計算時間は、小さな問題に対して 21.4 ~ 3936.9 秒、それ以外の問題に対して 31462.3 ~ 36000.0 秒となり、品種数 200 の問題に対してはすべての問題で計算時間の上限に達している。一方、*PPFE* の計算時間は、小さな問題に対して 12.9 ~ 721.4 秒、それ以外の問題に対して 2050.1 ~ 10019.3 秒となり、規模の大きな問題でも計算時間の上限に達していない。

表 6.11 は、個別の C 問題に対する計算時間である。- は実行可能解が求められていない問題、*t* は計算時間の上限に達した問題である。問題規模の小さな C1 ~ C13 に対して、閾値法の計算時間は 0.3 ~ 62.3 秒と小さい時間で解を算出しているが、実行可能解を算出できない場合も多い。容量スケールリング法の *PAFB* の計算時間は 0.8 ~ 172.6 秒、*PAFS* の計算時間は 1.0 ~ 737.1 秒であるが、*PAFE* の計算時間は 1.9 ~ 8164.9 秒と大きい。一方、*PPFE* の計算時間は 1.1 ~ 1854.4 秒であり、*PAFE* に比べて計算時間は小さい。

ノード数や品種数の多い規模の大きな C14 ~ C27 に対して、すべての解法の計算時間は大きい。これらの問題に対して、閾値法および *PAFE* では、計算時間が上限である 10 時間に達している場合が多い。閾値法では、C14 ~ C21 および C26, C27 のほとんどの定式化で計算時間の上限に達しており、C22 ~ C25 のほとんどの定式化で実行可能解を求

表 6.19 R 問題:規模別平均計算時間 *¹

N , A , K	閾値	アークフロー			パスフロー
	最大値	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	<i>PPFE</i>
10,35,10	0.1	0.6	0.8	2.1	0.9
10,35,25	0.3	3.9	5.8	16.9	6.6
10,35,50	0.9	8.5	12.1	49.9	16.1
10,60,10	0.1	0.9	1.2	2.4	0.9
10,60,25	1.2	10.0	19.0	51.7	18.0
10,60,50	6.4	73.4	62.4	391.0	78.9
10,83,10	0.2	2.9	4.3	13.0	4.1
10,83,25	1.6	8.5	16.0	152.0	32.1
10,83,50	10.0	131.6	79.3	297.3	76.2
20,120,40	55.7	813.8	1027.7	8577.0	746.8
20,120,100	813.4	2998.9	1837.9	12610.8	2203.5
20,120,200	1123.8	5117.5	4080.5	19745.7	3356.8
20,220,40	418.5	788.6	1309.5	5921.3	931.4
20,220,100	12106.9	11223.6	5600.0	22426.3	3952.2
20,220,200	23812.0	16118.9	20343.8	33617.1	7252.3
平均	2739.3	2664.1	2456.7	7418.0	1404.9

*1:閾値法で実行可能解が求められた 123 問

めることができていない。*PAFB*, *PAFS*, *PAFE* では, $C_{14} \sim C_{21}$, $C_{25} \sim C_{27}$ の計算時間が大きく, $C_{22} \sim C_{24}$ の計算時間が小さい。一方, *PPFE* の計算時間は, 最大で 21680.8 秒と計算時間の上限には達しておらず, 相対的には小さい。

R 問題の属性を表 6.12 および表 6.13 に示す。表の数値と記号は, ノード数, アーク数, 品種数, および区分の上限値と費用の性質である。R 問題では, 区分の上限値は三種類に分類され, c_1 は区分の上限値が大きい, c_8 は区分の上限値が小さいことを表す。また, 費用は三種類に分類され, f_{01} は固定費用が相対的に安い, f_{10} は相対的に高いことを表す。

表 6.14 は, R 問題を区分の上限値と費用の性質の組合せによって九つに分類し, 閾値法と容量スケールリング法の平均誤差を比較したものである。なお, 126 問の内, 閾値法では 3 問の実行可能解が求められていないため, 平均値は求められた 123 問に対する値である。閾値法の全体の平均誤差は 2.54% である。容量スケールリング法の全体の平均誤差は, *PAFB* で 3.54%, *PAFS* で 1.39%, *PAFE* で 1.13% であり, アークフローによる定式化を用いて算出した最良値の誤差は 1.03% である。また, *PPFE* の平均誤差は 1.23% である。*PPFE* の平均誤差は, *PAFE* の平均誤差よりも 0.10% 大きい。これは, *PPFE* におけるスケールリングの初期の繰り返しでは最適解または *PAFE* による最良解に含まれるパスフロー変数が十分に生成されていないため, 初期の繰り返しにおいて最適解または

PAFE による最良解に含まれるデザイン変数が 0 に収束していることが要因の一つと考えられる。

閾値法の性質別の誤差は、0.67~4.33% である。容量スケールリング法の誤差は、*PAFB* で 1.26~5.68%、*PAFS* で 0.43~2.18%、*PAFE* で 0.27~2.43% であり、アークフローによる定式化を用いて算出した最良値の計算誤差は 0.27~2.11% である。*PPFE* の誤差は 0.34~2.55% である。全体として、f01 問題の誤差は小さく、f10 問題の誤差が大きい。また、アークフローの最良値と *FFPE* は、すべての分類において閾値法より良い解を算出している。

表 6.15 は、R 問題をノード数、アーク数、品種数によって規模別に分類し、閾値法と容量スケールリング法の平均誤差を比較したものである。閾値法の規模別の誤差は、0.13~4.50% である。容量スケールリング法の誤差は、*PAFB* で 0.00~8.92%、*PAFS* で 0.00~5.34%、*PAFE* で 0.00~4.73% であり、最良値の誤差は 0.00~4.48% である。また、*PPFE* の誤差は 0.00~5.03% である。容量スケールリング法の誤差は規模が大きくなると増加するが、閾値法の誤差には顕著な傾向は見られない。ほぼすべての分類において容量スケールリング法が優れており、ノード数 10 の問題の誤差は 1% 以下、ノード数 20、アーク数 120 の問題の誤差は 1.3% 以下である。

表 6.16 および表 6.17 に、R 問題に対して性質別と規模別に最適解が求められた問題数を示す。閾値法により最適解が求められた問題数は 126 問の中の 10 問、容量スケールリング法の *PAFB* では 35 問、*PAFS* では 53 問、*PAFE* では 60 問、最良値では 67 問である。また、*PPFE* により最適解が求められた問題数は 53 問である。

表 6.18 は、R 問題に対する区分と費用の性質別の計算時間である。全体として、f 値が大きくなると計算時間が増加する傾向があるが、c 値に関しては共通した顕著な傾向は見られない。閾値法の計算時間は、c1、c2 問題では 2511.9~5386.8 秒と大きく、c8 問題では 183.6~698.0 秒と小さく、平均 2739.34 秒である。*PAFB* の計算時間は 1305.2~3395.0 秒で平均 2664.1 秒、*PAFS* の計算時間は 975.9~3738.7 秒で平均 2456.7 秒、*PAFE* の計算時間は 5648.7~10403.3 秒で平均 7418.0 秒となり、定式化が複雑になるにしたがって計算時間が増加している。一方、*PPFE* の計算時間は 466.7~2311.2 秒、平均 1404.9 秒である。*PPFE* の計算時間は、閾値法の 1/2、*PAFE* の 1/5.3 であり、列生成法により大幅に計算時間を短縮することができている。また、閾値法や *PAFE* ではいくつかの問題で計算時間の上限である 10 時間に達しているために計算を打ち切っており、実際の計算時間の差はさらに大きなものとなる。

表 6.19 は、R 問題に対する規模別の計算時間である。ノード数 10 の問題に対する計算時間が小さく、ノード数 20 の問題に対しては極端に計算時間が大きい。閾値法の計算時間は、ノード数 10 の問題に対して 0.1~10.0 秒、ノード数 20 の問題に対して 55.7~23812.0 秒である。*PAFB* の計算時間は、ノード数 10 の問題に対して 0.6~131.6 秒、ノード数 20 の問題に対して 788.6~16118.9 秒である。*PAFS* の計算時間は、ノード数 10 の問題に対して 0.8~79.3 秒、ノード数 20 の問題に対して 1027.7~20343.8 秒である。*PAFE* の計算時間は、ノード数 10 の問題に対して 2.1~391.0 秒、ノード数 20 の問題に

対して 8577.0 ~ 33617.1 秒である．一方，*PPFE* の計算時間は，ノード数 10 の問題に対して 0.9 ~ 78.9 秒，ノード数 20 の問題に対して 746.8 ~ 7252.3 秒であり，最大でも 2 時間程度である．

R 問題の個別の結果を付録の表 B.1 ~ B.5 に，個別の計算時間を表 B.6 ~ B.10 に示す．R 問題の 126 問の内，閾値法が良い解を算出しているのは 14 問のみであり，閾値法と容量スケール法とともに最適解が求められた問題が 10 問，それ以外の 102 問では容量スケール法が良い解を算出している．

6.6 本章のまとめ

本章では，区分的線形費用をもつネットワーク設計問題を対象として，アークフローによる定式化とパスフローによる定式化に対して，容量スケール法を用いた近似解法を提案した．これは，区分をダミーアークに置き換えた問題に容量スケール法を適用し，区分変数の範囲を制限した問題に対する分枝限定法を組み込んだ解法である．また，パスフローによる定式化に対して，列生成法と行生成法を用いた解法を示した．また，提案した解法に容量スケール法に対して数値実験を行い，提案した解法の有効性を示した．

本章では，区分的線形費用をもつネットワーク設計問題に対して，以下のような成果を示した．

- 1) 三種類のアークフローによる定式化とパスフローによる定式化を示した．

アークフローによる定式化は，基本的な定式化，アークと品種に関する需要の強制制約式を含む強い定式化，アーク，品種と区分に関する需要の強制制約式を含む拡張した定式化である．パスフローを用いた拡張した定式化は，アーク，品種と区分に関する需要の強制制約式を含む定式化である．また，区分線形モデルと多重アークモデルとの関係を示した．さらに，パスフローを用いた拡張した定式化の線形緩和問題において，被約費用と双対問題を示した．

- 2) 三種類の アークフローを用いた定式化に対して容量スケール法を示した．

容量の変更方法，容量スケール法の解をもとにした分枝限定法を示し，近似解法を含む容量スケール法を示した．

- 3) パスフローを用いた定式化に対して容量スケール法を示した．

容量の変更方法，線形緩和問題の限定主問題を示した．未生成の制約式に対する双対変数値の設定法を示し，限定主問題における価格付け問題が最短路問題となることを示した．また，行生成法と列生成法を示した．さらに，容量スケール法による解をもとにした分枝限定法を示し，近似解法を含む容量スケール法を示した．

- 4) 数値実験により，提案する解法の有効性を示した．

Crainic らのベンチマーク問題を用いて作成した C 問題と R 問題を用いて，アークフローを用いた三種類の定式化およびパスフローを用いた定式化に対する容量

スケーリング法と、従来の解法である閾値法を比較し、提案した解法の有効性を示した。閾値法では一部の問題で実行可能解を求めることができなかったが、容量スケーリング法はすべての問題で実行可能解を求めることができた。アークフローを用いた強い定式化に対する容量スケーリング法は、品種数の大きな問題では閾値法と同程度の誤差の解を得たが、それ以外の問題では閾値法よりも良い解を得ることができた。一方、パスフローを用いた強い定式化に対する容量スケーリング法では、計算時間を大幅に短縮することができたが、アークフローを用いた定式化と比べて解の精度は僅かながら劣るものとなった。

5) 問題の属性と適用すべき解法

品種数が大きな問題の場合は閾値法を適用し、それ以外の問題の場合は提案した *PAFE* を適用することが良いと考える。また、計算時間を抑えたい場合は、問題の属性によらず提案した *PPFE* を適用することが良いと考える。

今後の課題としては、次の項目が挙げられる。

1) 新たな妥当不等式の提示

品種・区分の需要に対する強制制約式を含めた定式化を用いたが、依然、問題によっては大きな双対ギャップが存在している。そのため、新たな有効な妥当不等式の開発が必要である。

2) アークフローを用いた定式化に対する容量スケーリング法の計算時間の短縮

アークフローを用いた定式化では、パスフローを用いた定式化による解法よりも、精度の高い解を求めることができたが、大きな計算時間を必要とする。そのため、不必要な変数や制約式の除去などによって、精度を低下させない範囲内で、計算時間の短縮を図る必要がある。

3) パスフローを用いた定式化に対する容量スケーリング法の解の精度の向上

パスフローを用いた定式化では、閾値法やアークフローを用いた定式化に比べて、大幅に計算時間を短縮することができたが、アークフローを用いた定式化と比べて解の精度は劣るものとなった。これは列を生成しながら、容量スケーリング法を行っていることが原因である。このため、解の精度を維持しながらのパスフローを用いた定式化に対する容量スケーリングを考える必要がある。

4) パスフローを用いた定式化に対する容量スケーリング法の計算時間

パスフローを用いた定式化では、大幅に計算時間を短縮することができたが、規模の大きな問題に対しては依然として大きな計算時間を必要とする。このために、不要な制約式や変数の除去による問題縮小やパラメータチューニングなどにより、さらなる計算時間の短縮を図る必要がある。

第7章

結論

7.1 まとめ

今日における急激な不況の進行とともに、多くの企業では国内や海外における生産拠点の開設、閉鎖および移転、物流拠点の統合や分散、輸送手段の選択や変更などといったロジスティクスネットワーク構造全体の再編が急務となってきている。しかし、多数の生産拠点、多数の物流拠点、複数の輸送手段、そして多数の品種など数多くの要因を考慮しなければならないため、数理的に適切なロジスティクスネットワークを設計することは、非常に困難なものとなっている。このようなロジスティクスネットワークを設計するための数理計画モデルとして、ロジスティクスネットワーク設計問題があり、今日まで大きな成果を挙げてきている。しかし、近年では、グローバル化とともにロジスティクスネットワークの規模が拡大していることから、大規模なロジスティクスネットワークに適用でき、かつ高速で精度の高いモデルと解法の開発が望まれている。

本論文では、このようなロジスティクスネットワーク設計に関連する基本的なネットワーク設計モデル、すなわち容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、および区分的線形費用をもつネットワーク設計問題を対象とした。これらの問題は、ロジスティクスネットワークを設計し、構築する際に発生する問題を抽象化したものであり、現実において考慮すべき様々な条件の中で基本的で重要な項目を考慮してモデル化したものである。また、数理面から最適に解くことが困難である *NP* 困難な問題のグループに属しており、実用規模の問題に対する一般的、効率的な最適解法は存在していないため、これらの問題に対する実用的な解を求める近似解法を開発することが課題となっている。

本論文では、はじめに、研究の背景、研究の目的、問題の提示および論文の構成を示し、ロジスティクスネットワーク設計問題に関する従来の研究を整理した。続いて、容量制約のないネットワーク設計問題に対して、強い妥当不等式の提示、大規模な問題に対しても適用が可能である高速な新しい近似解法を開発し、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発した。また、容量制約をもつネットワーク設計問題に対して、強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発した。さらに、

区分的線形費用をもつネットワーク設計問題に対して，強い定式化にもとづく精度の高い解を算出する新しい近似解法を開発した．

第1章

第1章では，研究の背景，研究の目的，問題の提示および論文の構成を示した．問題の提示では，ネットワークと最適化問題の項に分けて本論文に關係する用語の定義を示し，ネットワーク設計問題の定義を示し，その分類を解説した．さらに，ロジスティクスにおける応用問題の例を解説した．

第2章

第2章では，従来の研究の整理を行い，従来の研究における本論文の各章の位置づけを示した．従来の研究では，関連する問題を容量制約のないネットワーク設計問題，容量制約をもつネットワーク設計問題，区分的線形費用をもつネットワーク設計問題，応用問題，およびサーベイに分類し，解説した．

第3章

第3章では，容量制約のないネットワーク設計問題を取り扱い，この問題に対する強い妥当不等式といくつかの近似解法を示した．また，数値実験を行い，提案した解法の有効性を示した．

はじめに，最適解において，同一ノードを始点とする品種と終点とする品種のフローが同一のアーキ上で同じ向きに移動するフローが存在しない性質を用いて，新たな二種類の強い妥当不等式である非順流不等式を示した．続いて，非逆流不等式と非順流不等式を制約式として含むアーキフローによる定式化とパスフローによる定式化を示した．これらの定式化は強い妥当不等式を含むため強い定式化となり，緩和問題から求められる下界値が良いものとなる．しかし，非常に多くの制約を含むため，直接的に解くことは困難な問題となる．

次に，Minoux法をもとにした二種類の改良法を提案した．第一の改良法は，Minoux法におけるアーキ削除時の目的関数の評価に厳密性を加える改良を行った解法である．第二の改良法は，第一の方法に加え，アーキ削除時の目的関数の初期評価に厳密性を加える改良を行った解法である．これらの解法は，Minoux法に厳密性を加えているため，Minoux法よりも近似解の精度が高く，かつバックワード法より計算時間を短縮できることが期待できる．

次に，強い妥当不等式を含むパスフロー変数を用いた定式化の線形緩和問題を列生成法と行生成法を用いて解き，この線形緩和問題から近似解を導出する線形緩和法を示した．パスフロー変数を用いた定式化の線形緩和問題に対して，限定主問題，被約費用および価格付け問題を示し，価格付け問題が最短路問題に帰着されることを示した．続いて，線形緩和問題に対して，変数を追加する列生成法と制約式を追加する行生成法を示した．また，不要な変数と制約式を削除する問題の縮小方法を示した．さらに，線形緩和問題をもとにし

た限定した Minoux 法の改良法，限定した局所探索法と分枝限定法を用いた近似解法，および釘付けテストを示した．

最後に，数値実験により，提案した二種類の解法と従来の解法の比較を行い，提案した解法の有効性を示した．Minoux 法の改良法に対して，1000 ノード，5000 アーク，5000 品種までの問題について，数値実験を行い，従来の解法であるバックワード法，Minoux 法と比較した．Minoux 法の改良法は，バックワード法と同程度の精度の解を求めることができ，かつバックワード法に比べて大幅に計算時間を短縮することができた．線形緩和法に対して，100 ノード，500 アーク，500 品種までの問題について，数値実験を行い，バックワード法，Minoux 法，および CPLEX の分枝限定法と比較した．線形緩和法では，生成する列が多く，多くの計算時間を必要とするが，バックワード法，Minoux 法および CPLEX よりも良い上界値を求めることができた．

短時間で解を求めたい場合，ノード数 200，アーク数 1000 程度までの規模の問題に対してはバックワード法を適用し，ノード数 1000，アーク数 5000 程度までの規模の問題に対しては提案した Minoux の改良法を適用し，それ以上の規模の問題に対しては Minoux 法を適用することが良いと考える．また，精度の高い解を求めたい場合，ノード数 50，アーク数 250 程度までの規模の問題に対しては CPLEX を利用し，ノード数 100，アーク数 500 程度までの規模の問題に対しては提案した線形緩和法を適用し，ノード数 500，アーク数 2500 程度までの規模の問題に対してはバックワード法を適用し，それ以上の規模の問題に対しては提案した Minoux の改良法を適用することが良いと考える．

第 4 章

第 4 章では，第 3 章と同じ容量制約のないネットワーク設計問題を取り扱い，この問題に対する三種類の Lagrange 緩和法を示した．また，数値実験を行い，提案した解法の有効性を示した．

はじめに，三種類の定式化を示した．弱い定式化は，品種と需要に関する強制制約を含む定式化である．強い定式化は，始点を同じにする品種のフローに対する一種類の非逆流不等式とネットワークの連結条件を含む定式化である．拡張した定式化は，二種類の非逆流不等式，二種類の非順流不等式とネットワークの連結条件を含む定式化である．

次に，弱い定式化に対する Lagrange 緩和法を示した．フロー保存式を Lagrange 緩和した緩和問題を示し，この Lagrange 緩和問題がアーク毎の問題に分割できることを示した．さらに，分割された問題が係数の比較により容易に解けることを示し，緩和問題がデザイン変数のみを含む問題として表現できることを示した．また，Lagrange 乗数の変更法として，勾配法を示した．

次に，強い定式化に対する Lagrange 緩和法を示した．フロー保存式を Lagrange 緩和し，非逆流不等式を制約式とした緩和問題を示し，この Lagrange 緩和問題がアーク毎，始点毎の問題に分割することができることを示した．さらに，分割された問題が最小木問題に帰着されることを示し，緩和問題がデザイン変数のみを含む問題として表現できることを示した．また，Lagrange 乗数の変更法として，勾配法を示した．

次に、拡張した定式化に対する Lagrange 緩和法を示した。フロー保存式、二種類の非逆流のうち一種類と、二種類の非順流不等式を Lagrange 緩和し、一種類の非逆流不等式を制約式として残した Lagrange 緩和問題を示し、この Lagrange 緩和問題がアーク毎、始点毎の問題に分割することができることを示した。さらに、Lagrange 緩和問題が最小木問題に帰着されることを示し、緩和問題がデザイン変数のみを含む問題として表現できることを示した。また、Lagrange 乗数の変更法として、勾配法を示した。Lagrange 乗数の数が非常に多くなるため、必要に応じて Lagrange 乗数値を設定する方法を示した。

次に、Lagrange 緩和解をもとにした三種類の Lagrange ヒューリスティックを示した。Lagrange 緩和解において 1 であるデザイン変数、および緩和問題の目的関数の係数が小さい一定数のデザイン変数を初期解とし、この初期解をもとにした限定した Minoux タイプのフォワード法、限定した Minoux 法の改良法、限定した局所探索法を示した。

最後に、数値実験を行い、従来の解法である双対上昇法と比較した。下界値に対しては、いずれの Lagrange 緩和法も双対上昇法よりも良い値を求めることができた。特に、拡張した定式化では最も良い下界値を求めることができた。一方、上界値に対しては、いずれも同様な近似解法を用いているため、顕著な差は見られなかった。

きわめて短時間で下界値を求めたい場合は双対上昇法を適用し、それ以外の場合は提案した *UAFS* を適用することが良いと考える。また、より精度の高い下界値を求めたい場合は、提案した *UAFE* を適用することが良いと考える。

第5章

第5章では、容量制約のあるネットワーク設計問題を取り扱い、この問題の性質を検討し、三種類の定式化に対する容量スケールリング法を用いた近似解法を提案した。アークフロー変数を用いた定式化では、強制制約式を含まない弱い定式化と強制制約式を含む強い定式化について容量スケールリング法を適用した。パスフロー変数を用いた定式化では、強制制約式を含む強い定式化について容量スケールリング法を適用した。また、数値実験を行い、提案した解法の有効性を示した。

はじめに、アークフローを用いた強制制約式を含まない弱い定式化および強制制約式を含む強い定式化を示した。また、これらの定式化に対する二種類の線形緩和問題と二種類の Lagrange 緩和問題を示し、問題の性質と解法を示した。続いて、強制制約式を含むパスフローを用いた強い定式化を示した。また、この定式化の線形緩和問題を示し、線形緩和問題におけるパスフロー変数の被約費用と双対問題を示した。

次に、アークフローを用いた弱い定式化に対する容量スケールリング法および強い定式化に対する容量スケールリング法を示した。容量スケールリング法は、容量制約をもつ問題に対して、その線形緩和問題を解き、フロー解をもとに逐次容量を変更して、0-1 解を導出する方法である。また、容量スケールリング法によって得られた解と閾値を用いて近似解を求める近似解法を示した。これは、閾値以上のデザイン変数を 1、未満のデザイン変数を 0 と設定した多品種フロー問題を解くことによって近似解を算出する方法である。

次に、パスフローを用いた定式化に対して容量スケールリング法を示した。線形緩和問題

に対して、パスフロー変数を限定した限定主問題を示した。また、限定主問題における価格付け問題を示し、この問題が最短路問題となることを示した。続いて、逐次、パスフロー変数と強制制約式を付加する列生成法と行生成法を示した。さらに、容量スケールリング法によって得られた解をもとにした分枝限定法を示した。これは、閾値以上のデザイン変数を 1、もう一つの閾値以下のデザイン変数を 0 と設定した限定した分枝限定法を行うことによって近似解を算出する方法である。

最後に、Crainic らのベンチマーク問題を用いて数値実験を行い、単体法にもとづくタブー探索法、サイクルにもとづく近傍探索法、パス再結合法、多レベル共同タブー探索法と比較した。アークフローを用いた強い定式化に対する容量スケールリング法およびパスフローを用いた定式化に対する容量スケールリング法では、従来の解法に比べて多くの問題で良い解を求めることができた。パスフローを用いた定式化に対する容量スケールリング法では、列生成法により大幅に計算時間を短縮することができた。

ノード数が大きな問題に対しては多レベル共同タブー探索法を適用し、それ以外の問題に対しては提案した *CPFS* を適用することが良いと考える。また、短時間で解を求めたい場合は、問題の属性によらず提案した *CPFS* を適用することが良いと考える。

第 6 章

第 6 章では、区分的線形費用をもつネットワーク設計問題を取り扱い、この問題の性質を検討し、三種類のアークフローによる定式化に対する容量スケールリング法を用いた近似解法と、パスフローによる定式化に対する容量スケールリング法を用いた近似解法を示した。また、数値実験を行い、提案した解法の有効性を示した。

はじめに、三種類のアークフローによる定式化とパスフローによる定式化を示した。アークフローによる定式化は、基本的な定式化、アークと品種に関する需要の強制制約式を含む強い定式化、アーク、品種と区分に関する需要の強制制約式を含む拡張した定式化である。パスフローを用いた拡張した定式化は、アーク、品種と区分に関する需要の強制制約式を含む定式化である。また、区分線形モデルと多重アークモデルとの関係を示した。さらに、パスフローを用いた拡張した定式化の線形緩和問題において、被約費用と双対問題を示した。

次に、三種類のアークフローを用いた定式化に対して、容量スケールリング法、および容量スケールリング法による解をもとにした分枝限定法を示した。容量スケールリング法は、容量制約をもつ問題に対して、その線形緩和問題を解き、フロー解をもとに逐次容量を変更して、0-1 解を導出する方法である。多区分を多重アークに置き換えることによって、区分的線形費用をもつネットワーク設計問題を容量制約をもつ問題に変換し、容量スケールリング法を適用可能にした。また、近似解法として、閾値以下のデザイン変数を 0 と設定した限定した分枝限定法により近似解を算出する方法を示した。

次に、パスフローを用いた定式化に対して、容量スケールリング法、およびパスフロー変数を限定した限定主問題を示した。未生成の制約式に対する双対変数値の設定法を示し、限定主問題における価格付け問題が最短路問題となることを示した。続いて、逐次、パス

フロー変数と強制制約式を付加する列生成法と行生成法を示した。さらに、容量スケールリング法による解をもとにした分枝限定法を示し、近似解法を含む容量スケールリング法を示した。

最後に、Crainicらのベンチマーク問題をもとに作成した問題を用いて数値実験を行い、アークフローを用いた三種類の定式化とパスフローを用いた定式化に対する容量スケールリング法と従来の解法である閾値法を比較した。

閾値法が実行可能解を求めることができない場合があるのに対して、容量スケールリング法はすべての問題において実行可能解を求めることができた。アークフローを用いた強い定式化に対する容量スケールリング法では、品種数の大きな問題に対して閾値法と同程度の誤差の解を得たが、それ以外の問題に対して閾値法よりも良い解を得ることができた。一方、パスフローフローを用いた強い定式化に対する容量スケールリング法も、品種数の大きな問題に対して閾値法と同程度の誤差の解を得たが、それ以外の問題に対して閾値法よりも良い解を得ることができた。さらに、計算時間を大幅に短縮することができたが、アークフローを用いた定式化と比べて解の精度は僅かながら劣るものとなった。

品種数が大きな問題に対しては閾値法を適用し、それ以外の問題に対しては提案した *PAFE* を適用することが良いと考える。また、計算時間を抑えたい場合は、問題の属性によらず提案した *PPFE* を適用することが良いと考える。

7.2 今後の課題

本論文では、容量制約のないネットワーク設計問題、容量制約をもつネットワーク設計問題、および区分的線形費用をもつネットワーク設計問題に関して、従来の研究の整理といくつかの解法の開発を行い、多くの成果を挙げたが、今後の課題も数多く残されている。

第3章

新たな妥当不等式である二種類の非順流不等式を提案し、ノード数が少ない問題では線形緩和が元の問題の最適解に一致することも多く、有効な妥当不等式であることが示された。しかし、規模が大きな問題では、非順流不等式を満たす非 0-1 の緩和が数多く生成され、その値も 0 または 1 から離れたものも多くなる。そのため、より強力な妥当不等式を提示することが課題である。

Minoux 法の改良法を提案した。この解法は、大規模な問題に非常に高速に適用できる解法である。そのため、タブー探索法などの繰り返し型のメタ解法を組み込むことが可能である。また、アーク削除に制限を加えることによって、輸送距離や輸送時間の制限、ノード経由回数の制限などの実際的な付帯制約をもつ問題に拡張できる可能性がある。

列生成と行生成法を用いた線形緩和法を提案したが、容量制約のない問題では、非常に多くのパスと制約式が生成され、多くの計算時間が必要となる。このため、さらなる不要なパスの削除、不要なパスを生成しないなどによる問題規模の縮小化が必要である。また、分枝限定法に移行する際の自由変数の数の適切な制御方法や、分枝限定法への列生成

法と行生成法の組み込みが必要である。また、パスを変数として扱っていることから、生成するパスを限定することによって、輸送距離や輸送時間の制限、ノード経由回数の制限などの実際的な付帯制約をもつ問題に拡張できる可能性がある。

第4章

第4章では、三種類の Lagrange 緩和法を示した。強い定式化と拡張した定式化による下界値は大変優れたものであるが、ノード数が大きくなると誤差が大きくなっている。このため、規模の大きな問題における誤差の改善が課題であり、新たな妥当不等式の考慮や有効な Lagrange ヒューリスティックの開発が必要となる。

また、ノード数が増加すると計算時間が膨大なものとなる。特に、Lagrange ヒューリスティックの計算時間が大きくなる。このため、解の精度を大きく変えない範囲で、Lagrange ヒューリスティックの実施回数の適切な制御や Lagrange ヒューリスティック自体の高速化などが必要である。

また、特に拡張した定式化による Lagrange 緩和法では、パラメータが多く存在するため、パラメータの適切なチューニングが必要である。

第5章

第5章では、三種類の容量スケールリング法を用いた近似解法を提案した。パスフローを用いた定式化に対する容量スケールリング法により、従来の解法から得られる解よりも多くの問題で優れた解を求めることができた。しかし、ノード数が大きな問題に関しては、従来の解法が優れている場合があり、これらの要因の解析と解法へのフィードバックが必要である。また、品種・区分の需要に対する強制制約式を含めた定式化を用いたが、問題によっては大きな双対ギャップが存在するため、新たな有効な妥当不等式の開発が必要である。

また、列生成法と行生成法を組合せたパスフローに対する定式化による解法は、アークフローによる定式化による解法に比べて計算時間を大幅に短縮することができた。しかし、さらに規模の大きな問題を対象とする場合には、不要なパスや強制制約式の削除による問題の縮小などによる高速化が必要である。また、分枝限定法への列生成法と行生成法の組み込みが必要である。

第6章

第6章では、アークフローによる定式化に対する三種類の容量スケールリング法を用いた近似解法と、パスフローによる定式化に対する容量スケールリング法を用いた近似解法を示した。

品種・区分の需要に対する強制制約式を含めた定式化を用いたが、問題によっては大きな双対ギャップが存在するため、新たな有効な妥当不等式の開発が必要である。また、アークフローを用いた定式化に対する容量スケールリング法により多くの問題で良い解を求めることができた。しかし、一部の問題では従来の解法に劣る場合もあるため、その要因

を調べ、解法にフィードバックすることによる解法の改良が必要である。また、十分なパラメータのチューニングも必要である。

規模の大きな問題に対するアークフローによる定式化に対する容量スケールリング法では、多くの計算時間を必要とするため、十分なスケールリングの繰り返し回数を確保することができなかった。そのため、計算時間の短縮が必要である。

パスフローを用いた定式化に対する容量スケールリング法では、アークフローによる定式化に比べて大幅に計算時間を短縮することができたが、アークフローを用いた定式化と比べて解の精度は劣るものとなった。このため、解の精度を維持しながらの容量スケールリングを検討する必要がある。また、規模の大きな問題に対しては依然として大きな計算時間を必要とするために、さらなる計算時間の短縮を図る必要がある。

全体

本論文では、アークに上で発生する費用を最小化するようなアークの選択によるネットワークの形状とモノの移動経路を求める基本的なロジスティクスネットワーク設計問題を対象とした。これらの設計問題は、必ずしも現実のロジスティクスにおけるネットワーク設計における条件をすべて考慮している訳ではないが、ネットワーク設計に関する根幹的な条件を考慮している。しかし、これらのモデルや解法を現実のロジスティクスネットワーク設計問題に適用するためには多くの条件や前提を加え、それに伴うモデルの変更や解法の改良が必要となる。また、現実のデータに対応するための改良も必要となる。

現実のロジスティクスネットワーク設計問題に適用するにあたり、取り入れるべき条件としては、例えば、ノード設計、多期間モデル、在庫モデル、排出ガス量などの環境条件、輸送機器のフロー保存モデル、結束条件、輸送サービスレベル、フローの木条件、積替回数条件、為替・関税条件などがある。設計の際に発生する問題として、例えば、設備配置問題、ロットサイズ問題、生産スケールリング問題、配送計画問題、在庫配置問題、在庫方策選択などがある。これらの条件のいくつかはロジスティクスネットワーク設計モデルに付加され、また単独の問題として、従来から数多く研究されている。

ノード設計は、工場や倉庫の配置の選択であり、ノードをアークに変換することによりある程度対応することが可能であるが、規模が拡大する。多期間モデルは、ネットワークを期間毎に作成し、これらのネットワークを時間推移を表すアークで連結することにより対応することが可能であるが、ネットワークの規模が大幅に拡大する。また、在庫を考慮する場合、非線形関数である凹関数の要素が加わり、区分的線形関数に近似することによりある程度対応することが可能であるが、規模は大幅に拡大する。その他のいくつかの条件は、分解法や緩和法などの解法を組合せて対応することが必要となる。いずれにしても、現実のロジスティクスネットワーク設計問題は複雑な要素を含み、その規模は大きく、直接、全体を最適化するように解くことは困難である。

本論文で提案した基本的な問題に対する解法は、現実的なロジスティクスネットワーク設計問題の解法の基礎として、また解法の一部に組込むなどして利用することが可能である。今後の展開としては、本論文で対象とした基本的な問題に現実問題で優先的に考慮す

べき条件を付加したモデルを作成し，本論文で提案した解法をモデルに合うように改良することが考えられる．また，現実の問題に適用する際には，現実の問題を基本的な設計問題に集約し，この基本的な設計問題と提案した解法を用いて解を求め，この解をもとにロジスティクスネットワーク全体の構造を決定し，続いてこのネットワークの構造を基本として，さらなる詳細な条件を考慮した下位の問題に分解し，下位の問題を個別に解くことにより全体の最適化を目指すことが考えられる．

以上のように，本論文による研究成果を従来からの研究成果を融合し，より現実的な問題に適用しうる解法を開発することが課題となる．また，現実の問題に適用できる統合的な解析システムを開発し提供し，事例への適用していくことも視野に入れるべきである．

参考文献

- [1] R. K. Ahuja and V. V. S. Murty. New lower planes for the network design problem. *Networks*, Vol. 17, pp. 113–127, 1987.
- [2] A. M. Alvarez, J. L. González-Velarde, and K. De-Alba. Grasp embedded scatter search for the multicommodity capacitated network design problem. *Journal of Heuristics*, Vol. 11, pp. 233–257, 2005.
- [3] A. M. Alvarez, J. L. González-Velarde, and K. De-Alba. Scatter search for network design problem. *Annals of Operations Research*, Vol. 138, No. 1, pp. 159–178, 2005.
- [4] B. C. Arntzen, G. C. Brown, T. P. Harrion, and L. L. Trafton. Global supply chain management at digital equipment corporation. *Interfaces*, Vol. 25, pp. 69–93, 1995.
- [5] A. Balakrishnan. LP extreme points and cuts for the fixed-charge network design problem. *Mathematical Programming*, Vol. 39, pp. 263–284, 1987.
- [6] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. Network design. In M. Dell’Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pp. 311–334. John Wiley & Sons, New York, 1997.
- [7] A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, Vol. 37, pp. 716–740, 1989.
- [8] F. Barahona. Network design using cut inequalities. *SIAM journal on Computing*, Vol. 6, pp. 823–837, 1996.
- [9] J. W. Billheimer and P. Gray. Network design with fixed and variable cost elements. *Transportation Science*, Vol. 7, pp. 49–74, 1973.
- [10] T. B. Boffey and A. I. Hinxman. Solving the optimal network problem. *European journal of Operational Research*, Vol. 3, pp. 386–393, 1979.
- [11] D. E. Boyce, A. Farhi, and R. Weischedel. Optimal network problem : Branch-and-bound algorithm. *Environment and Planning*, Vol. 5, pp. 519–533, 1973.
- [12] G. G. Brown, G. W. Graves, and M. D. Honczarenko. Design and operation of a multicommodity production/distribution system using primal goal decomposi-

- tion. *Management Science*, Vol. 33, pp. 1469–1480, 1987.
- [13] J. F. Campbell. Hub location and the p-hub median problem. *Operations Research*, Vol. 44, pp. 923–935, 1996.
- [14] M. Chouman, T. G. Crainic, and B. Gendron. A cutting-plane algorithm based on cutset inequalities for multicommodity capacitated fixed charge network design. Technical Report CRT-2003-16, Centre de recherche sur les transports, Université de Montréal, 2003.
- [15] A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research*, Vol. 32, pp. 1429 – 1450, 2005.
- [16] T. G. Crainic. Handbook of transportation science. In R. W. Hall, editor, *Long-haul Freight Transportation*, pp. 451–516. Kluwer Academic Publishers, 2003.
- [17] T. G. Crainic, A. Frangioni A., and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems. *Discrete Applied Mathematics*, Vol. 112, pp. 73–99, 2001.
- [18] T. G. Crainic, M. Gendreau, and J. M. Farvolden. A simplex-based tabu search for capacitated network design. *INFORMS journal on Computing*, Vol. 12, pp. 223–236, 2000.
- [19] T. G. Crainic and B. Gendron. Cooperative parallel tabu search for capacitated network design. *journal of Heuristics*, Vol. 8, pp. 601–627, 2002.
- [20] T. G. Crainic, B. Gendron, and G. Hernu. A slope scaling/Lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, Vol. 10, pp. 525–545, 2004.
- [21] T. G. Crainic, Y. Li, and M. Toulouse. A first multilevel cooperative algorithm for capacitated multicommodity network design. *Computers & Operations Research*, Vol. 33, pp. 2602–2622, 2006.
- [22] K. L. Croxton, B. Gendron, and T. L. Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, Vol. 49, pp. 1268–1273, 2003.
- [23] K. L. Croxton, B. Gendron, and T. L. Magnanti. Models and methods for merge-in-transit operations. *Transportation Science*, Vol. 37, pp. 1–22, 2003.
- [24] K. L. Croxton, B. Gendron, and T. L. Magnanti. Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research*, Vol. 55, pp. 146–157, 2007.
- [25] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.
- [26] R. Dionne and M. Florian. Exact and approximate algorithms for optimal network design. *Networks*, Vol. 9, pp. 37–59, 1979.

-
- [27] A. T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, Vol. 4, pp. 139–154, 1996.
- [28] A. T. Ernst and M. Krishnamoorthy. An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS journal on Computing*, Vol. 10, pp. 149–162, 1998.
- [29] G. Gallo. Lower planes for the network design problem. *Networks*, Vol. 13, pp. 411–425, 1983.
- [30] B. Gendron. A note on "a dual-ascent approach to the fixed-charge capacitated network design problems". Technical Report CRT-99-38, Centre de recherche sur les transports, Université de Montréal, 1999.
- [31] B. Gendron and T. G. Crainic. Parallel implementations of bounding procedures for multicommodity capacitated network design. Technical Report CRT-94-45, Centre de recherche sur les transports, Université de Montréal, 1994.
- [32] B. Gendron and T. G. Crainic. Relaxations for multicommodity capacitated network design problems. Technical Report CRT-96-5, Centre de recherche sur les transports, Université de Montréal, 1994.
- [33] B. Gendron and T. G. Crainic. Bounding procedures for multicommodity capacitated fixed charge network design problems. Technical Report CRT-96-06, Centre de recherche sur les transports, Université de Montréal, 1996.
- [34] B. Gendron, T. G. Crainic, and A. Frangioni. Multicommodity capacitated network design. Technical Report CRT-98-14, Centre de recherche sur les transports, Université de Montréal, 1997.
- [35] A. M. Geoffrion and G. Graves. Multicommodity distribution system design by benders' decomposition. *Management Science*, Vol. 5, pp. 822–844, 1974.
- [36] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, Vol. 51, pp. 655–667, 2003.
- [37] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Path relinking, cycle-based neighborhoods and capacitated multicommodity network design. *Annals of Operations Research*, Vol. 131, pp. 109–134, 2004.
- [38] M. Goetschalck, C. J. Vidal, and K. Dogan. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*, Vol. 143, pp. 1–18, 2002.
- [39] G. J. Guie'erez, P. Kouvelis, and A. A. Kurawarwala. A robustness approach to uncapacitated network design problems. *European journal of Operational Research*, Vol. 94, pp. 362–376, 1996.
- [40] J. W. Herrmann, G. Ioannou, I. Minis, and J. M. Proth. A dual ascent approach

- to the fixed-charge capacitated network design problem. *European journal of Operational Research*, Vol. 95, pp. 476–490, 1996.
- [41] H. H. Hoang. A computational approach to the selection of an optimal network. *Management Science*, Vol. 19, pp. 488–498, 1973.
- [42] K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a Lagrangean heuristic and branch-and-bound. *Operations Research*, Vol. 46, pp. 247–259, 1998.
- [43] K. Holmberg and D. Yuan. A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, Vol. 48, pp. 461–481, 2000.
- [44] D. S. Johnson, J. K. Lenstra, and A. H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, Vol. 8, pp. 279–285, 1978.
- [45] 片山直登, 春日井博. 容量制約をもつ多品種流ネットワークデザイン問題. 日本経営工学会誌, Vol. 44, pp. 164–175, 1993.
- [46] 片山直登, 春日井博. ラグランジュ緩和法を用いた予算制約をもつネットワークデザイン問題の解法. 日本経営工学会誌, Vol. 46, pp. 21–27, 1995.
- [47] 片山直登. ネットワークデザイン問題. 久保幹雄, 田村明久, 松井知己 (編), 応用数理計画ハンドブック, pp. 1184–1282. 朝倉書店, 2002.
- [48] 片山直登. ネットワーク設計問題. 朝倉書店, 2008.
- [49] D. Kim, X. Pan, and P. M. Pardalos. An enhanced dynamic slope scaling procedure with tabu scheme for fixed charge network flow problems. *Computational Economics*, Vol. 27, pp. 273–293, 2006.
- [50] D. Kim and P. M. Pardalos. A dynamic domain contraction algorithm for non-convex piecewise linear network flow problems. *journal of Global Optimization*, Vol. 17, pp. 225–234, 2000.
- [51] D. Kim and P. M. Pardalos. Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems. *Networks*, Vol. 35, pp. 216–222, 2000.
- [52] J. G. Klincewicz. Heuristics for the p-hub location problem. *European journal of Operational Research*, Vol. 53, pp. 25–37, 1991.
- [53] J.B. Kruskal. On the shortest spanning tree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, pp. 48–50, 1956.
- [54] 久保幹雄, 田村明久, 松井知己. 応用数理計画ハンドブック. 朝倉書店, 2002.
- [55] B. W. Lamar, Y. Sheffi, and W. R. Powell. A capacity improvement lower bound for fixed charge network design problems. *Operations Research*, Vol. 38, pp. 704–710, 1990.
- [56] M. Los and C. Lardinois. Combinatorial programming, statistical optimization and the optimal transportation network problem. *Transportation Research B*,

- Vol. 16, pp. 89–124, 1982.
- [57] T. L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, Vol. 60, pp. 233–250, 1993.
- [58] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring benders decomposition for uncapacitated network design. *Mathematical Programming Study*, Vol. 26, pp. 112–155, 1986.
- [59] T. L. Magnanti and R. T. Wong. Network design and transportation planning : Models and algorithms. *Transportation Science*, Vol. 18, pp. 1–55, 1984.
- [60] M. Minoux. Multiflots de coût minimal avec fonctions de coût concaves. *Annales des Télécommunications*, Vol. 31, pp. 77–92, 1976.
- [61] M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, Vol. 19, pp. 313–360, 1989.
- [62] 森津秀夫. 最適交通網構成手法に関する基礎的手法. 神戸大学, 1984.
- [63] J. D. Murchland. *A fixed matrix method for all shortest distances in a directed graph and for the inverse problem*. Ph. D. Thesis, University of Karlsruhe, 1970.
- [64] A. Muriel and F. N. Munshi. Capacitated multicommodity network flows with piecewise linear concave costs. *IIE Transactions*, Vol. 36, pp. 683–696, 2004.
- [65] M. E. O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European journal of Operational Research*, Vol. 32, pp. 393–404, 1987.
- [66] Y. Pochet and M. V. Vyve. A general heuristic for production planning problems. *journal on Computing*, Vol. 16, pp. 316 – 327, 2004.
- [67] W. B. Powell. A local improvement heuristic for the design of less-than-truckload motor carrier networks. *Transportation Science*, Vol. 20, pp. 246–257, 1986.
- [68] W. B. Powell and Y. Sheffi. Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Operations Research*, Vol. 37, pp. 12–29, 1989.
- [69] J. Roy and T. G. Crainic. Improving intercity freight routing with a tactical planning model. *Interfaces*, Vol. 22, pp. 31–44, 1992.
- [70] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, Vol. 167, pp. 96–115, 2005.
- [71] A. J. Scott. The optimal network problem: Some computational procedures. *Transportation Research*, Vol. 3, pp. 201–210, 1969.
- [72] D. Skorin-Kapov, J. Skorin-Kapov, and M. E. O’Kelly. On tabu search for the location of interacting hub facilities. *European journal of Operational Research*, Vol. 73, pp. 502–509, 1994.
- [73] P. A. Steenbrink. *Optimization of Transport Networks*. John Wiley & Sons, New

- York, 1974.
- [74] R. T. Wong. Worst-case analysis of network design problem heuristics. *SIAM journal on Algebra Discrete Mathematics*, Vol. 1, pp. 51–63, 1980.
- [75] R. T. Wong. Introduction and recent advances in network design: Models and algorithms. In M. Florian, editor, *Transportation Planning Models*, pp. 187–225. Elsevier Science, North Holland, Amsterdam, 1984.
- [76] R. T. Wong. Location and network design. In M. O’heEigartaigh, J. Lenstra, and A. RinnooyKan, editors, *Combinatorial Optimization Annotated Bibliographies*, pp. 129–147. John Wiley & Sons, New York, 1985.
- [77] R. T. Wong. Probabilistic analysis of a network design problem heuristic. *Networks*, Vol. 15, pp. 347–363, 1985.
- [78] N. C. Zaleta and A. M. A. Socarrás. Tabu search-based algorithm for capacitated multicommodity network design problems. *14th International Conference on Electronics, Communications and Computers*, pp. 144 – 148, 2004.

付録 A

表 A.1 *CND* R 問題:個別結果 1

問題	最適値 または 下界値	従来法		<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
		上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R01-1	74079 ^O	74079	0.00	74079	0.00	74079	0.00	74079	0.00	0.00
R01-2	92403 ^O	92403	0.00	92403	0.00	92403	0.00	92403	0.00	0.00
R01-3	115304 ^O	115304	0.00	115304	0.00	115304	0.00	115304	0.00	0.00
R01-4	84908 ^O	84908	0.33	85169	0.31	84931	0.03	84931	0.03	0.03
R01-5	113036 ^O	113036	1.33	114565	1.35	114565	1.35	114565	1.35	1.35
R01-6	147599 ^O	147599	2.11	150821	2.18	150821	2.18	150821	2.18	2.18
R02-1	232239 ^O	232239	0.00	232239	0.00	232239	0.00	232239	0.00	0.00
R02-2	322453 ^O	326333	0.99	325493	0.94	323861	0.44	323861	0.44	-0.76
R02-3	419503 ^O	424512	3.33	433614	3.36	420018	0.12	419503	0.00	-1.18
R02-4	316437 ^O	316437	0.00	316437	0.00	316437	0.00	316437	0.00	0.00
R02-5	431250 ^O	431889	0.00	431250	0.00	432224	0.23	432224	0.23	0.08
R02-9	559578 ^O	563570	1.22	566732	1.28	559578	0.00	559578	0.00	-0.71
R03-1	484830 ^O	484830	0.00	484830	0.00	484830	0.00	484830	0.00	0.00
R03-2	703362 ^O	710840	0.22	704801	0.20	706676	0.47	706676	0.47	-0.59
R03-3	944990 ^O	965330	0.44	949227	0.45	944990	0.00	944990	0.00	-2.11
R03-4	704247 ^O	704247	0.00	704247	0.00	704247	0.00	704247	0.00	0.00
R03-5	932897 ^O	932897	0.00	932897	0.00	932897	0.00	932897	0.00	0.00
R03-6	1188638 ^O	1188796	0.22	1191057	0.20	1190999	0.20	1190999	0.20	0.19

O:最適値

表 A.2 *CND* R 問題:個別結果 2

問題	最適値 または 下界値	従来法 による 上界値	<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R04-1	31730 ^O	31730	31730	0.00	31730	0.00	31730	0.00	0.00
R04-2	48920 ^O	48920	49159	0.49	48920	0.00	48920	0.00	0.00
R04-3	63767 ^O	63767	63767	0.00	63767	0.00	63767	0.00	0.00
R04-4	33740 ^O	33740	33970	0.68	33886	0.43	33760	0.06	0.06
R04-5	53790 ^O	53790	55424	3.04	53956	0.31	54296	0.94	0.94
R04-6	74030 ^O	74030	77212	4.30	77426	4.59	75109	1.46	1.46
R04-7	68292 ^O	68293	68292	0.00	68292	0.00	68292	0.00	0.00
R04-8	113004 ^O	113226	113226	0.20	113226	0.20	113226	0.20	0.00
R04-9	163208 ^O	164430	167008	2.33	168074	2.98	166453	1.99	1.23
R05-1	123003 ^O	123003	125567	2.08	123003	0.00	123003	0.00	0.00
R05-2	170060 ^O	170467	170467	0.24	170467	0.24	170060	0.00	-0.24
R05-3	221486 ^O	221486	232165	4.82	225713	1.91	222904	0.64	0.64
R05-4	131608 ^O	131608	131877	0.20	131722	0.09	131722	0.09	0.09
R05-5	204157 ^O	205764	210694	3.20	207208	1.49	204593	0.21	-0.57
R05-6	286524 ^O	292244	294553	2.80	296286	3.41	292341	2.03	0.03
R05-7	278372 ^O	278372	278372	0.00	278372	0.00	278372	0.00	0.00
R05-8	445810 ^O	449477	445913	0.02	445913	0.02	445913	0.02	-0.79
R05-9	625879 ^O	629040	630075	0.67	626291	0.07	626291	0.07	-0.44
R06-1	245936 ^O	248394	250083	1.69	245936	0.00	245936	0.00	-0.99
R06-2	401685 ^O	412283	428720	6.73	405979	1.07	401685	0.00	-2.57
R04-3	559477 ^O	578752	596307	6.58	559477	0.00	559477	0.00	-3.33
R06-4	286682 ^O	288460	288748	0.72	286873	0.07	287580	0.31	-0.31
R06-5	498266 ^O	515967	529148	6.20	502517	0.85	502676	0.89	-2.58
R06-6	734414 ^O	771683	802143	9.22	738896	0.61	738896	0.61	-4.25
R06-7	682921 ^O	682921	682921	0.00	682921	0.00	682921	0.00	0.00
R06-8	1030479 ^O	1042968	1030923	0.04	1030479	0.00	1030479	0.00	-1.20
R06-9	423316 ^O	438860	467385	10.41	424622	0.31	423688	0.09	-3.46

O:最適値

表 A.3 *CND* R 問題:個別結果 3

問題	最適値 または 下界値	従来法 による 上界値	<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R07-1	32807 ^O	32807	32807	0.00	32807	0.00	32807	0.00	0.00
R07-2	47252 ^O	47252	47252	0.00	47252	0.00	47252	0.00	0.00
R07-3	62962 ^O	62962	64731	2.81	62962	0.00	62962	0.00	0.00
R07-4	37432 ^O	37432	37432	0.00	37432	0.00	37432	0.00	0.00
R07-5	56475 ^O	56591	56720	0.43	56747	0.48	56915	0.78	0.57
R07-6	77249 ^O	78875	79246	2.59	78375	1.46	78375	1.46	-0.63
R07-7	59947 ^O	59947	60222	0.46	60222	0.46	60111	0.27	0.27
R07-8	99194 ^O	100155	102214	3.04	103334	4.17	100663	1.48	0.51
R07-9	141692 ^O	142319	146367	3.30	146747	3.57	146560	3.44	2.98
R08-1	102531 ^O	102556	104038	1.47	102531	0.00	102645	0.11	0.09
R08-2	143894 ^O	143894	157258	9.29	143894	0.00	143894	0.00	0.00
R08-3	182793 ^O	182793	195296	6.84	182793	0.00	182793	0.00	0.00
R08-4	109325 ^O	109325	109958	0.58	109325	0.00	109325	0.00	0.00
R08-5	157047 ^O	158168	166094	5.76	157720	0.43	157720	0.43	-0.28
R08-6	207540 ^O	208135	218752	5.40	208135	0.29	208325	0.38	0.09
R08-7	154160 ^O	155384	156892	1.77	154316	0.10	155836	1.09	0.29
R08-8	274867 ^O	283133	280564	2.07	284771	3.60	280730	2.13	-0.85
R08-9	415793 ^O	429896	438474	5.45	432059	3.91	426900	2.67	-0.70
R09-1	171512 ^O	172343	174127	1.52	171902	0.23	171923	0.24	-0.24
R09-2	296712 ^O	307038	323953	9.18	298155	0.49	296712	0.00	-3.36
R09-3	424266 ^O	435590	474099	11.75	425978	0.40	424266	0.00	-2.60
R09-4	192736 ^O	193242	196120	1.76	193144	0.21	192833	0.05	-0.21
R09-5	357318 ^O	371998	379523	6.21	363892	1.84	357318	0.00	-3.95
R09-6	522187 ^O	555945	571329	9.41	525338	0.60	526455	0.82	-5.30
R09-7	345057 ^O	348297	345184	0.04	345728	0.19	345728	0.19	-0.74
R09-8	646579 ^O	662147	653079	1.01	652143	0.86	651186	0.71	-1.66
R09-9	951136 ^O	987938	959227	0.85	975016	2.51	962719	1.22	-2.55

O:最適値; L:下界値

表 A.4 *CND* R 問題:個別結果 4

問題	最適値 または 下界値	従来法 による 上界値	<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R10-1	200087 ^O	200613	202196	1.05	200157	0.03	200087	0.00	-0.26
R10-2	346814 ^O	350573	383263	10.51	352041	1.51	348998	0.63	-0.45
R10-3	488015 ^O	507118	569155	16.63	492676	0.96	497406	1.92	-1.92
R10-4	229196 ^O	231546	233428	1.85	230976	0.78	229849	0.28	-0.73
R10-5	411664 ^O	432913	446664	8.50	418186	1.58	418396	1.64	-3.35
R10-6	609104 ^O	640621	670867	10.14	632802	3.89	609104	0.00	-4.92
R10-7	486895 ^O	488737	488341	0.30	487667	0.16	487014	0.02	-0.35
R10-8	951056 ^O	980010	960941	1.04	960574	1.00	959154	0.85	-2.13
R10-9	1421740 ^O	1454329	1443441	1.53	1428653	0.49	1426459	0.33	-1.92
R11-1	714431 ^O	725416	721952	1.05	714431	0.00	714431	0.00	-1.51
R11-2	1263713 ^O	1306090	1380808	9.27	1298629	2.76	1264665	0.08	-3.17
R11-3	1843611 ^O	1914040	2014047	9.24	1923573	4.34	1856995	0.73	-2.98
R11-4	870451 ^O	876894	880219	1.12	871324	0.10	871275	0.09	-0.64
R11-5	1623640 ^O	1694860	1690048	4.09	1649356	1.58	1625443	0.11	-4.10
R11-6	2414060 ^O	2607690	2510729	4.00	2447036	1.37	2430141	0.67	-6.81
R11-7	2294912 ^O	2295790	2294912	0.00	2295439	0.02	2295439	0.02	-0.02
R11-8	3507100 ^O	3568430	3514610	0.21	3509272	0.06	3508387	0.04	-1.68
R11-9	4579353 ^O	4621900	4595101	0.34	4579353	0.00	4579353	0.00	-0.92
R12-1	1639443 ^O	1713670	1679137	2.42	1642488	0.19	1640889	0.09	-4.25
R12-2	3360268 ^L	3746250	3643891	8.44	3514124	4.58	3427145	1.99	-8.52
R12-3	5144559 ^L	6070200	5650189	9.83	5401707	5.00	5313090	3.28	-12.47
R12-4	2303557 ^O	2326230	2306448	0.13	2306448	0.13	2303557	0.00	-0.97
R12-5	4669799 ^O	4950073	4732602	1.34	4693655	0.51	4669799	0.00	-5.66
R12-6	7100019 ^O	7638050	7235229	1.90	7126938	0.38	7101247	0.02	-7.03
R12-7	7635270 ^O	7637250	7635270	0.00	7635270	0.00	7635270	0.00	-0.03
R12-8	10067742 ^O	10121700	10067742	0.00	10067742	0.00	10067742	0.00	-0.53
R12-9	11967768 ^O	12079300	11980075	0.10	11967768	0.00	11967768	0.00	-0.92

O:最適値; L:下界値

表 A.5 *CND* R 問題:個別結果 5

問題	最適値 または 下界値	従来法 による 上界値	<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R13-1	142947 ^O	144138	146638	2.58	142947	0.00	142947	0.00	-0.45
R13-2	263800 ^O	270316	299233	13.43	269709	2.24	265179	0.52	-1.90
R13-3	365836 ^O	374999	419206	14.59	375094	2.53	370229	1.20	-1.27
R13-4	150977 ^O	151513	154808	2.54	150977	0.00	151471	0.33	-0.03
R13-5	282682 ^O	291510	316288	11.89	293608	3.87	284283	0.57	-2.48
R13-6	406790 ^O	420028	450984	10.86	428885	5.43	412045	1.29	-1.90
R13-7	208088 ^O	212451	219576	5.52	209910	0.88	210676	1.24	-0.84
R13-8	441944 ^L	484112	473512	7.14	468315	5.97	465614	5.36	-3.82
R13-9	686486 ^L	758715	763982	11.29	730731	6.45	730213	6.37	-3.76
R14-1	403414 ^O	415119	431248	6.90	403414	0.00	403703	0.07	-2.75
R14-2	749429 ^L	803356	862070	15.03	757434	1.07	754726	0.71	-6.05
R14-3	1040970 ^L	1155840	1235444	18.68	1103328	5.99	1074037	3.18	-7.08
R14-4	437607 ^O	453204	460421	5.21	438644	0.24	438370	0.17	-3.22
R14-5	839290 ^L	912456	940920	12.11	865015	3.07	853937	1.75	-6.41
R14-6	1194757 ^L	1333440	1301623	8.94	1227646	2.75	1216473	1.82	-8.77
R14-7	665247 ^L	702226	685784	3.09	672664	1.11	668692	0.52	-4.78
R14-8	1587532 ^L	1748930	1734623	9.27	1665364	4.90	1632060	2.80	-6.68
R14-9	2577261 ^L	2882710	2769387	7.45	2666325	3.46	2670985	3.64	-7.34
R15-1	1000787 ^O	1049360	1061865	6.10	1002143	0.14	1000787	0.00	-4.63
R15-2	1912558 ^L	2158720	2237985	17.02	1985436	3.81	1978546	3.45	-8.35
R15-3	2738164 ^L	3135760	3147974	14.97	2956787	7.98	2907820	6.20	-7.27
R15-4	1146858 ^L	1215130	1195457	4.24	1151623	0.42	1149298	0.21	-5.42
R15-5	2410765 ^L	2756680	2698435	11.93	2499080	3.66	2482469	2.97	-9.95
R15-6	3696226 ^L	4384640	4166779	12.73	4016765	8.67	3859004	4.40	-11.99
R15-7	2297691 ^L	2355730	2304306	0.29	2311719	0.61	2302234	0.20	-2.27
R15-8	5573413 ^O	5926330	5615869	0.76	5587946	0.26	5583203	0.18	-4.38
R15-9	8696932 ^O	9180920	8755687	0.68	8696932	0.00	8696932	0.00	-5.27

O:最適値; L:下界値

表 A.6 *CND* R 問題:個別結果 6

問題	最適値 または 下界値	従来法 による 上界値	<i>CAFW</i>		<i>CAFS</i>		<i>CPFS</i>		
			上界値	誤差 (%)	上界値	誤差 (%)	上界値	誤差 (%)	改善率 (%)
R16-1	136161 ^O	136538	142667	4.78	136161	0.00	136161	0.00	-0.28
R16-2	239500 ^O	247682	270304	12.86	243150	1.52	240221	0.30	-3.01
R16-3	325671 ^O	338807	372744	14.45	342165	5.06	325839	0.05	-3.83
R16-4	138532 ^O	139973	145920	5.33	139038	0.37	138532	0.00	-0.55
R16-5	241801 ^O	246014	275019	13.74	244059	0.93	241801	0.00	-1.71
R16-6	337762 ^O	355610	396599	17.42	344997	2.14	342618	1.44	-3.65
R16-7	168951 ^L	172268	174001	2.99	172918	2.35	173721	2.82	0.84
R16-8	339516 ^L	365214	371892	9.54	367610	8.27	359918	6.01	-1.45
R16-9	509101 ^L	569874	589937	15.88	548615	7.76	540807	6.23	-5.10
R17-1	354138 ^O	370090	382633	8.05	354223	0.02	354223	0.02	-4.29
R17-2	643636 ^L	688554	759951	18.07	672496	4.48	652404	1.36	-5.25
R17-3	874666 ^L	971151	1075596	22.97	948315	8.42	921465	5.35	-5.12
R17-4	370590 ^O	380850	389593	5.13	371144	0.15	370622	0.01	-2.69
R17-5	698457 ^L	753188	830836	18.95	714169	2.25	713227	2.11	-5.31
R17-6	973365 ^L	1108180	1183946	21.63	1075709	10.51	1032134	6.04	-6.86
R17-7	497323 ^L	524038	519026	4.36	513059	3.16	505029	1.55	-3.63
R17-8	1079856 ^L	1195140	1184616	9.70	1111765	2.95	1111641	2.94	-6.99
R17-9	1719164 ^L	1945080	1962407	14.15	1851345	7.69	1802089	4.82	-7.35
R18-1	828034 ^L	872888	904351	9.22	837194	1.11	830776	0.33	-4.82
R18-2	1500315 ^L	1716680	1804187	20.25	1554789	3.63	1544308	2.93	-10.04
R18-3	2076541 ^L	2377560	2494437	20.12	2246240	8.17	2211771	6.51	-6.97
R18-4	912710 ^L	975396	985362	7.96	931695	2.08	923609	1.19	-5.31
R18-5	1761083 ^L	2037950	1983197	12.61	1869383	6.15	1838483	4.40	-9.79
R18-6	2577531 ^L	2966370	2952774	14.56	3097986	20.19	2734439	6.09	-7.82
R18-7	1458573 ^L	1622520	1557953	6.81	1506342	3.28	1483337	1.70	-8.58
R18-8	3776540 ^L	4576750	4064524	7.63	4200721	11.23	3927531	4.00	-11.84
R18-9	6143618 ^L	7125415	6735473	9.63	7058801	14.90	6415624	4.43	-9.96

O:最適値; L:下界値

表 A.7 CND R 問題:個別計算時間 1

問題	サイクル ル(秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)	問題	サイクル ル(秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
R01-1	8.7	0.0	0.0	0.0	R04-1	11.5	0.1	0.0	0.0
R01-2	9.0	0.1	0.0	0.0	R04-2	13.4	0.1	0.0	0.0
R01-3	8.9	0.1	0.1	0.1	R04-3	12.3	0.2	0.0	0.0
R01-4	9.6	0.2	0.1	0.1	R04-4	13.9	0.2	0.2	0.1
R01-5	11.8	0.1	0.2	0.1	R04-5	14.5	0.1	0.2	0.2
R01-6	11.5	0.1	0.2	0.1	R04-6	14.0	0.1	0.3	0.1
					R04-7	18.9	0.1	0.2	0.1
					R04-8	18.3	0.1	0.2	0.1
					R04-9	18.3	0.2	0.2	0.2
R02-1	19.3	0.1	0.1	0.1	R05-1	30.4	0.2	0.1	0.1
R02-2	26.0	0.2	0.2	0.1	R05-2	30.0	0.3	0.3	0.1
R02-3	26.1	0.2	0.3	0.1	R05-3	29.4	0.3	0.6	0.3
R02-4	27.6	0.2	0.2	0.1	R05-4	33.5	0.3	0.3	0.1
R02-5	34.9	0.2	0.2	0.1	R05-5	35.1	0.3	0.5	0.2
R02-6	29.9	0.1	0.2	0.1	R05-6	36.5	0.3	0.8	0.8
					R05-7	63.3	0.2	0.3	0.2
					R05-8	66.4	0.3	0.4	0.1
					R05-9	69.1	0.3	0.5	0.1
R03-1	47.7	0.3	0.2	0.1	R06-1	56.4	0.5	0.5	0.1
R03-2	55.2	0.3	0.4	0.2	R06-2	66.3	0.5	1.0	0.3
R03-3	59.3	0.3	0.4	0.1	R04-3	66.2	0.5	2.2	0.2
R03-4	59.6	0.3	0.1	0.0	R06-4	69.6	0.5	0.7	0.6
R03-5	59.6	0.2	0.4	0.1	R06-5	80.8	0.5	1.6	2.0
R03-6	73.5	0.2	0.4	0.0	R06-6	79.7	0.4	4.2	1.1
					R06-7	155.2	0.1	0.4	0.1
					R06-8	153.1	0.1	0.2	0.1
					R06-9	55.5	0.4	1.8	0.6

表 A.8 CND R 問題:個別計算時 2

問題	サイクル ル (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)	問題	サイクル ル (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
R07-1	15.7	0.0	0.0	0.0	R10-1	124.2	0.6	0.9	0.3
R07-2	15.3	0.0	0.0	0.0	R10-2	135.1	0.7	2.1	2.3
R07-3	16.3	0.1	0.3	0.1	R10-3	161.9	0.9	7.8	9.5
R07-4	17.0	0.1	0.2	0.1	R10-4	80.3	0.8	1.4	0.4
R07-5	16.9	0.2	0.3	0.1	R10-5	196.4	0.7	3.0	1.6
R07-6	17.0	0.2	0.3	0.1	R10-6	175.4	0.7	8.6	3.1
R07-7	21.8	0.2	0.3	0.2	R10-7	464.7	1.3	2.4	1.1
R07-8	21.6	0.2	0.3	0.2	R10-8	414.6	1.5	3.4	1.0
R07-9	20.6	0.3	0.3	0.3	R10-9	332.9	1.0	4.8	1.1
R08-1	39.1	0.3	0.3	0.1	R11-1	392.6	2.0	5.3	0.8
R08-2	36.8	0.3	0.1	0.1	R11-2	631.1	5.0	70.8	33.6
R08-3	43.0	0.3	0.1	0.1	R11-3	674.6	3.7	141.8	28.4
R08-4	38.4	0.3	0.3	0.1	R11-4	580.6	8.9	15.3	1.8
R08-5	49.0	0.4	0.6	0.2	R11-5	1069.8	5.6	29.0	7.6
R08-6	49.5	0.4	1.1	0.6	R11-6	1244.4	6.4	55.2	11.4
R08-7	64.3	0.4	0.6	0.3	R11-7	1126.3	2.4	3.3	0.9
R08-8	49.2	0.4	1.0	0.6	R11-8	1160.2	2.9	6.3	0.8
R08-9	74.1	0.3	1.2	1.3	R11-9	786.5	2.9	2.0	1.1
R09-1	73.1	0.5	0.7	0.1	R12-1	3012.1	11.5	25.8	3.3
R09-2	76.1	0.6	1.7	1.5	R12-2	3274.9	18.5	158.0	158.7
R09-3	84.1	0.7	2.2	1.6	R12-3	3612.7	28.8	377.2	96.7
R09-4	89.7	0.7	1.0	0.2	R12-4	4700.4	18.6	53.9	3.3
R09-5	104.2	0.7	2.5	2.0	R12-5	7932.1	14.3	64.2	4.8
R09-6	119.6	0.6	3.1	0.2	R12-6	4728.5	11.9	40.5	5.2
R09-7	29.7	0.7	1.4	0.2	R12-7	4728.3	2.3	4.9	0.9
R09-8	136.9	0.8	1.8	0.5	R12-8	2978.3	1.9	2.3	0.9
R09-9	165.3	0.8	1.7	1.4	R12-9	2284.0	1.8	2.1	0.3

表 A.9 CND R 問題:個別計算時間 3

問題	サイクル ル (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)	問題	サイク ル (秒)	CAFW (秒)	CAFS (秒)	CPFS (秒)
R13-1	277.4	1.1	1.5	0.2	R16-1	404.9	1.6	0.2	0.1
R13-2	274.7	1.3	12.3	5.2	R16-2	416.9	2.0	18.0	10.7
R13-3	259.2	1.3	28.2	11.2	R16-3	148.5	2.3	66.4	63.6
R13-4	252.3	1.1	2.1	0.3	R16-4	376.8	1.6	2.5	0.3
R13-5	285.3	1.4	14.6	6.1	R16-5	430.9	1.8	43.1	6.7
R13-6	310.8	1.9	79.2	16.9	R16-6	421.6	1.9	53.1	27.8
R13-7	393.5	1.5	5.5	1.1	R16-7	477.1	1.9	6.4	1.7
R13-8	483.3	2.8	53.7	12.0	R16-8	520.9	3.3	105.0	8.0
R13-9	504.5	4.5	83.2	11.1	R16-9	577.9	4.6	90.7	178.1
R14-1	615.2	2.8	7.4	1.4	R17-1	909.2	3.8	12.4	1.9
R14-2	822.8	3.1	156.4	199.6	R17-2	998.6	4.9	456.8	56.2
R14-3	765.6	3.6	307.5	121.0	R17-3	999.4	8.0	1570.3	126.0
R14-4	665.4	3.0	9.0	1.7	R17-4	927.3	4.3	24.2	2.7
R14-5	971.3	5.1	405.8	46.9	R17-5	1070.0	6.1	368.8	64.3
R14-6	862.7	7.2	311.1	75.2	R17-6	1168.9	9.4	1701.9	157.8
R14-7	1814.6	14.7	36.6	16.9	R17-7	1315.7	6.8	335.6	14.6
R14-8	3793.7	33.3	154.5	23.6	R17-8	2508.8	29.9	561.2	219.8
R14-9	3858.4	29.6	274.7	22.1	R17-9	2813.2	29.6	1325.7	135.9
R15-1	1767.1	8.6	97.3	4.0	R18-1	2339.0	9.2	187.0	17.4
R15-2	2521.4	16.6	2122.2	257.6	R18-2	2889.6	19.7	4423.2	531.1
R15-3	2618.4	35.1	4119.5	382.9	R18-3	2918.6	20.8	6041.0	747.4
R15-4	2794.5	23.3	210.1	105.3	R18-4	2677.5	13.2	1251.7	36.0
R15-5	5980.8	82.5	1650.2	525.6	R18-5	4335.1	70.2	7800.8	947.3
R15-6	5133.2	91.5	3949.3	271.9	R18-6	4234.0	134.3	1198.8	762.4
R15-7	15947.9	86.6	258.4	11.3	R18-7	13514.9	198.6	177.0	399.8
R15-8	13398.5	50.4	127.2	6.4	R18-8	28883.1	75.9	200.9	71.1
R15-9	12288.1	46.8	113.3	6.0	R18-9	32695.2	113.3	312.3	96.8

付録 B

表 B.1 PND R 問題:個別結果 1

問題	最適値/ 下界値	閾値		アークフロー				パスフロー	
		最良値	誤差	PAFB	PAFS	PAFE	誤差	PPFE	誤差
R01-1	56167 ^O	56167	0.00	56167	56167	56167	0.00	56167	0.00
R01-2	74491 ^O	75556	1.43	74491	74491	74491	0.00	74491	0.00
R01-3	96643 ^O	96643	0.00	97392	96643	96643	0.00	96643	0.00
R01-4	62751 ^O	64138	2.21	62751	62751	62751	0.00	62751	0.00
R01-5	90752 ^O	93861	3.43	90752	90752	90752	0.00	90752	0.00
R01-6	125016 ^O	131485	5.17	127645	126689	125016	0.00	125016	0.00
R02-1	195335 ^O	195501	0.08	195335	195335	195335	0.00	195335	0.00
R02-2	281371 ^O	289178	2.77	281371	281371	281371	0.00	281371	0.00
R02-3	378061 ^O	378061	0.00	378061	378061	378061	0.00	378061	0.00
R02-4	281309 ^O	283035	0.61	281309	281309	281309	0.00	281309	0.00
R02-5	393453 ^O	405856	3.15	393453	393453	393453	0.00	393453	0.00
R02-9	518169 ^O	530799	2.44	518169	518169	518169	0.00	518169	0.00
R03-1	417256 ^O	418456	0.29	417256	417256	417256	0.00	417256	0.00
R03-2	629630 ^O	629630	0.00	629630	629630	629630	0.00	629630	0.00
R03-3	876326 ^O	878100	0.20	876326	876326	876326	0.00	876326	0.00
R03-4	618763 ^O	619117	0.06	618763	618763	618763	0.00	618763	0.00
R03-5	843169 ^O	845094	0.23	843169	843169	843169	0.00	843169	0.00
R03-6	1074572 ^O	1074572	0.00	1074572	1074572	1074572	0.00	1074572	0.00

O:最適値

表 B.2 PND R 問題:個別結果 2

問題	最適値/ 下界値	閾値		アークフロー				パスフロー	
		最良値	誤差	PAFB	PAFS	PAFE	誤差	PPFE	誤差
R04-1	26620 ^O	26620	0.00	26864	26620	26620	0.00	26620	0.00
R04-2	42579 ^O	42579	0.00	43253	42579	42579	0.00	42579	0.00
R04-3	56297 ^O	56297	0.00	58605	56297	56297	0.00	56297	0.00
R04-4	26661 ^O	—	—	26743	26743	26661	0.00	26661	0.00
R04-5	45311 ^O	47510	4.85	49130	45408	45408	0.21	45619	0.68
R04-6	63204 ^O	69645	10.19	69875	64441	64441	1.96	64441	1.96
R04-7	54962 ^O	55750	1.43	55118	55118	55118	0.28	55118	0.28
R04-8	98540 ^O	100751	2.24	99462	99817	100228	0.94	99874	1.35
R04-9	148014 ^O	—	—	148014	148014	149828	0.00	152979	3.35
R05-1	96004 ^O	96311	0.32	97110	96004	96143	0.00	96143	0.14
R05-2	138009 ^O	141961	2.86	139949	138009	138009	0.00	138009	0.00
R05-3	180853 ^O	194609	7.61	194350	180853	180853	0.00	180853	0.00
R05-4	97431 ^O	97485	0.06	97431	97431	97431	0.00	97431	0.00
R05-5	165013 ^O	171832	4.13	168372	166189	165013	0.00	166324	0.79
R05-6	237210 ^O	251253	5.92	243704	237210	237210	0.00	237210	0.00
R05-7	236122 ^O	266445	12.84	236166	236122	236122	0.00	236122	0.00
R05-8	399040 ^O	410519	2.88	399040	399040	399665	0.00	399040	0.00
R05-9	557560 ^O	575885	3.29	557560	557560	564878	0.00	564878	1.31
R06-1	194813 ^O	195528	0.37	194813	194813	194813	0.00	196413	0.82
R06-2	328923 ^O	334060	1.56	330557	328923	328923	0.00	331593	0.81
R06-3	476370 ^O	482194	1.22	482866	477020	477020	0.14	476370	0.00
R06-4	219907 ^O	220561	0.30	222613	220331	219907	0.00	220896	0.45
R06-5	423660 ^O	425376	0.41	437240	425376	423660	0.00	423660	0.00
R06-6	657493 ^O	659712	0.34	694723	674266	667640	1.54	662888	0.82
R06-7	578776 ^O	703343	21.52	578776	578776	578776	0.00	578776	0.00
R06-8	905550 ^O	905550	0.00	905550	905550	905550	0.00	905550	0.00
R06-9	357262 ^O	357654	0.11	357262	357262	357262	0.00	358941	0.47

O:最適値

表 B.3 PND R 問題:個別結果 3

問題	最適値/ 下界値	閾値		アークフロー				パスフロー	
		最良値	誤差	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	誤差	<i>PPFE</i>	誤差
R07-1	27626 ^O	27626	0.00	27626	27626	27626	0.00	27626	0.00
R07-2	41110 ^O	43416	5.61	41114	41110	41110	0.00	41110	0.00
R07-3	55053 ^O	56904	3.36	56930	55053	55053	0.00	55053	0.00
R07-4	29122 ^O	30751	5.59	29122	29122	29122	0.00	29122	0.00
R07-5	47406 ^O	50189	5.87	48428	47905	47624	0.46	47624	0.46
R07-6	66119 ^O	73355	10.94	70146	66598	66598	0.72	66982	1.31
R07-7	44710 ^O	45235	1.17	44965	45219	44825	0.26	44710	0.00
R07-8	82054 ^O	82756	0.86	88088	86762	83037	1.20	83798	2.13
R07-9	123942 ^O	129656	4.61	130819	128437	127577	2.93	128437	3.63
R08-1	84236 ^O	84809	0.68	84372	84236	84236	0.00	84236	0.00
R08-2	119036 ^O	120125	0.91	119036	119036	119036	0.00	119036	0.00
R08-3	147238 ^O	—	—	155874	147238	147238	0.00	147238	0.00
R08-4	82920 ^O	83477	0.67	83976	83796	82920	0.00	82935	0.02
R08-5	124013 ^O	130783	5.46	131589	126267	126202	1.77	125667	1.33
R08-6	170789 ^O	175744	2.90	175718	172954	173051	1.27	174829	2.37
R08-7	119300 ^O	122456	2.65	120535	120156	119582	0.24	119736	0.37
R08-8	240354 ^O	244820	1.86	255102	244051	243306	1.23	243306	1.23
R08-9	380626 ^O	387724	1.86	398254	390171	381218	0.16	385960	1.40
R09-1	136413 ^O	136671	0.19	137877	136413	136656	0.00	137136	0.53
R09-2	247909 ^O	249336	0.58	250048	248688	248688	0.31	247909	0.00
R09-3	359230 ^O	361202	0.55	359230	359230	359230	0.00	359230	0.00
R09-4	149778 ^O	150269	0.33	152337	149778	149778	0.00	149864	0.06
R09-5	295459 ^O	297567	0.71	305444	295459	295459	0.00	295459	0.00
R09-6	439484 ^O	443152	0.83	455133	441021	439484	0.00	440186	0.16
R09-7	297705 ^O	298928	0.41	300028	298776	297756	0.02	297756	0.02
R09-8	605063 ^O	608022	0.49	605415	605857	605063	0.00	605415	0.06
R09-9	905366 ^O	933782	3.14	909790	905957	905585	0.02	905585	0.02

O:最適値

表 B.4 PND R 問題:個別結果 4

問題	最適値/ 下界値	閾値		アークフロー				パスフロー	
		最良値	誤差	<i>PAFB</i>	<i>PAFS</i>	<i>PAFE</i>	誤差	<i>PPFE</i>	誤差
R10-1	159900 ^O	162660	1.73	163742	160395	159900	0.00	159900	0.00
R10-2	291116 ^O	296519	1.86	319879	292331	294251	0.42	295966	1.67
R10-3	421206 ^O	433076	2.82	487884	421206	430694	0.00	430694	2.25
R10-4	178175 ^O	179498	0.74	179523	179309	178639	0.26	178652	0.27
R10-5	343650 ^O	344252	0.18	381454	344028	344028	0.11	348778	1.49
R10-6	532164 ^O	546221	2.64	581731	544750	540632	1.59	547449	2.87
R10-7	428713 ^O	453780	5.85	429263	428713	428713	0.00	428868	0.04
R10-8	896043 ^O	931170	3.92	903664	901486	900053	0.45	900024	0.44
R10-9	1358312 ^O	1421959	4.69	1363227	1358312	1359463	0.00	1360287	0.15
R11-1	548720 ^O	552272	0.65	555970	550320	548720	0.00	548720	0.00
R11-2	1036774 ^O	1067855	3.00	1108526	1081686	1061815	2.42	1062015	2.43
R11-3	1564101 ^O	1626456	3.99	1697293	1640979	1631395	4.30	1625605	3.93
R11-4	707652 ^O	713446	0.82	719602	714209	710459	0.40	710518	0.40
R11-5	1446789 ^O	1456355	0.66	1509131	1465039	1456234	0.65	1456234	0.65
R11-6	2210582 ^O	2277625	3.03	2343105	2281821	2272411	2.80	2259602	2.22
R11-7	1989904 ^O	1991542	0.08	1989904	1989904	1989904	0.00	1989904	0.00
R11-8	3096263 ^O	3112464	0.52	3096263	3096263	3096263	0.00	3096263	0.00
R11-9	4001315 ^O	4007277	0.15	4001315	4001315	4007272	0.00	4001587	0.01
R12-1	1292985 ^O	1299747	0.52	1320481	1300353	1299672	0.52	1298286	0.41
R12-2	2925656 ^O	3014062	3.02	3164912	3065530	3055151	4.43	3010445	2.90
R12-3	4635961 ^O	4919939	6.13	5075312	4996227	5013757	7.77	5000881	7.87
R12-4	2056033 ^O	2059883	0.19	2071684	2061145	2056033	0.00	2056033	0.00
R12-5	4468906 ^O	4486730	0.40	4481607	4504196	4468906	0.00	4468906	0.00
R12-6	6908255 ^O	6947220	0.56	6923101	6923101	6923101	0.21	6923101	0.21
R12-7	6158973 ^O	6169887	0.18	6158973	6158973	6158973	0.00	6158973	0.00
R12-8	7979079 ^O	7997409	0.23	7979079	7979079	7979079	0.00	7979079	0.00
R12-9	9286859 ^O	9291317	0.05	9286859	9286859	9286859	0.00	9286859	0.00

O:最適値

表 B.5 PND R 問題:個別結果 5

問題	最適値/ 下界値	閾値		アークフロー				パスフロー	
		最良値	誤差	PAFB	PAFS	PAFE	誤差	PPFE	誤差
R13-1	123681 ^O	124520	0.68	123681	124674	123681	0.00	123681	0.00
R13-2	227175 ^O	230639	1.52	241914	232964	228343	0.51	229571	1.05
R13-3	319612 ^O	323783	1.31	344649	322830	321739	0.67	325194	1.75
R13-4	120899 ^O	124565	3.03	129527	121295	120899	0.00	121264	0.30
R13-5	234750 ^O	241870	3.03	264001	246703	237750	1.28	242021	3.10
R13-6	341366 ^O	364609	6.81	388099	360185	355860	4.25	357230	4.65
R13-7	160336 ^O	166084	3.58	178468	166575	164791	2.78	165101	2.97
R13-8	385089 ^O	397100	3.12	424297	401838	394266	2.38	394182	2.36
R13-9	624964 ^O	650853	4.14	695046	662823	656380	5.03	659658	5.55
R14-1	328111 ^O	332478	1.33	342560	331211	331124	0.92	331124	0.92
R14-2	586082 ^O	637495	8.77	663630	641429	633810	8.14	638074	8.87
R14-3	825714 ^O	907220	9.87	950371	914929	922310	10.80	938253	13.63
R14-4	341502 ^O	342346	0.25	360491	344183	342740	0.36	343385	0.55
R14-5	668007 ^O	698921	4.63	718234	698117	700294	4.51	699128	4.66
R14-6	985361 ^O	1038482	5.39	1067017	1033176	1034850	4.85	1043510	5.90
R14-7	526556 ^O	535936	1.78	555290	539702	535520	1.70	535520	1.70
R14-8	1430403 ^O	1480344	3.49	1569570	1542542	1490395	4.19	1480595	3.51
R14-9	2393296 ^O	2512489	4.98	2583584	2545408	2529565	5.69	2525859	5.54
R15-1	758258 ^L	782508	3.20	798555	784508	783228	3.29	783656	3.35
R15-2	1528015 ^O	1620372	6.04	1677650	1617911	1647467	5.88	1633046	6.87
R15-3	2297266 ^O	2428229	5.70	2555422	2432620	2452704	5.89	2439750	6.20
R15-4	872881 ^O	895596	2.60	916524	896605	896504	2.71	896751	2.73
R15-5	2019166 ^O	2083360	3.18	2290131	2089997	2094764	3.51	2090320	3.52
R15-6	3265119 ^L	3408356	4.39	3727736	3400683	3454296	4.15	3415080	4.59
R15-7	2074464 ^L	2084878	0.50	2102954	2099642	2085841	0.55	2082064	0.37
R15-8	5402469 ^L	5431449	0.54	5406828	5424166	5409670	0.08	5408317	0.11
R15-9	8395444 ^L	8498092	1.22	8473990	8475870	8475720	0.94	8469114	0.88

O:最適値; L:下界値

表 B.6 PND R 問題:個別計算時間 1

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
R01-1	—	—	0.1	0.3	0.3	0.8	0.4
R01-2	—	—	0.1	0.4	0.2	1.1	0.7
R01-3	—	—	0.1	0.4	0.4	1.1	0.9
R01-4	—	0.1	0.1	0.9	1.0	1.8	1.2
R01-5	—	0.1	0.1	0.9	1.1	1.5	0.8
R01-6	—	0.1	0.1	0.7	1.6	6.3	1.6
R02-1	—	—	0.2	7.4	11.6	13.2	4.8
R02-2	0.1	0.1	0.3	5.0	5.0	9.8	3.1
R02-3	0.1	0.2	0.4	4.1	5.7	15.6	6.5
R02-4	0.1	—	0.3	1.8	4.9	12.7	3.5
R02-5	0.1	—	0.3	2.9	4.8	25.6	9.9
R02-9	0.1	—	0.4	2.4	2.9	24.6	11.9
R03-1	0.2	—	0.8	10.7	13.7	30.9	11.1
R03-2	0.3	—	0.9	9.2	14.7	34.6	14.6
R03-3	0.2	—	1.7	14.6	27.5	136.9	44.0
R03-4	0.1	—	0.7	6.8	4.3	49.5	12.9
R03-5	0.1	—	0.6	6.2	7.7	36.1	11.5
R03-6	0.1	—	0.6	3.6	4.6	11.8	2.6

—:実行可能解なし

表 B.7 PND R 問題:個別計算時間 2

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
R04-1	—	—	0.1	0.4	0.2	0.1	0.1
R04-2	—	—	0.1	0.3	0.3	0.1	0.1
R04-3	—	—	0.1	0.6	0.2	0.4	0.5
R04-4	—	—	—	0.5	0.7	2.7	0.8
R04-5	—	0.1	0.1	0.6	1.5	2.2	0.8
R04-6	—	0.1	0.1	0.6	1.1	2.6	0.9
R04-7	—	—	0.1	1.6	2.4	3.9	1.8
R04-8	—	—	0.2	1.8	2.2	4.9	1.6
R04-9	—	—	—	2.1	1.7	4.8	1.6
R05-1	0.1	0.2	0.5	5.3	3.2	7.2	1.9
R05-2	0.2	0.9	1.0	4.4	38.1	40.2	17.5
R05-3	—	1.2	1.2	3.5	27.2	94.4	19.2
R05-4	—	0.3	0.4	6.0	9.2	16.6	6.1
R05-5	—	1.6	0.8	10.1	34.0	44.0	10.7
R05-6	—	1.3	0.9	30.1	10.5	30.1	8.8
R05-7	0.1	—	1.3	12.0	13.8	82.8	42.4
R05-8	0.1	—	1.6	9.8	15.3	106.5	46.6
R05-9	0.1	—	1.8	9.1	19.8	43.6	9.1
R06-1	—	1.0	2.0	67.8	14.8	141.6	26.7
R06-2	1.3	4.5	16.0	85.3	56.1	224.2	59.1
R06-3	1.5	7.2	7.3	77.2	118.8	363.8	97.1
R06-4	—	2.7	2.5	23.7	56.1	364.6	37.6
R06-5	0.7	8.7	4.4	97.9	64.0	965.8	64.7
R06-6	0.9	11.6	11.4	160.8	196.8	1204.0	278.8
R06-7	0.2	—	2.9	4.4	7.0	52.9	57.3
R06-8	0.2	0.6	2.7	10.3	20.7	80.5	45.5
R06-9	3.6	2.7	3.4	132.9	27.7	122.0	43.3

—:実行可能解なし

表 B.8 PND R 問題:個別計算時間 3

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
R07-1	—	—	0.1	0.2	0.3	0.5	0.3
R07-2	—	—	0.2	0.8	1.4	1.5	1.2
R07-3	—	—	0.3	1.1	0.9	1.5	1.5
R07-4	—	0.1	—	1.0	1.3	2.9	0.8
R07-5	—	0.1	0.2	0.9	1.8	2.9	1.5
R07-6	—	0.1	0.2	1.1	3.0	3.8	4.5
R07-7	—	0.1	0.2	9.0	12.7	21.2	2.1
R07-8	—	0.3	0.2	9.6	11.6	56.0	16.5
R07-9	—	—	0.4	2.4	5.7	26.5	9.0
R08-1	0.5	—	0.4	4.9	5.4	6.2	1.8
R08-2	—	—	0.6	5.2	3.9	3.9	2.2
R08-3	—	—	—	3.1	1.6	2.4	1.2
R08-4	0.2	—	0.5	8.0	4.3	16.7	3.1
R08-5	—	0.6	1.1	6.3	19.5	41.6	11.1
R08-6	—	0.5	1.1	9.2	25.4	51.4	20.5
R08-7	—	1.0	1.4	8.6	21.4	402.7	27.7
R08-8	—	1.4	5.6	12.4	22.6	487.9	76.2
R08-9	0.1	1.5	3.3	18.9	40.0	355.6	144.8
R09-1	2.4	0.6	1.4	95.6	34.8	61.2	13.9
R09-2	25.8	2.2	4.3	411.7	69.0	212.3	82.5
R09-3	18.1	29.7	27.8	119.0	210.1	544.9	273.0
R09-4	3.8	1.4	1.6	75.5	92.5	231.5	30.6
R09-5	9.8	3.9	9.1	175.0	132.2	582.9	104.6
R09-6	—	6.1	5.7	196.4	82.6	290.7	88.6
R09-7	—	1.1	2.4	24.9	26.4	113.9	15.9
R09-8	0.4	—	5.3	27.8	29.6	293.6	16.6
R09-9	0.2	—	4.5	58.8	36.5	345.1	60.6

—:実行可能解なし

表 B.9 PND R 問題:個別計算時間 4

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
R10-1	—	—	4.5	45.7	47.8	90.3	9.4
R10-2	—	10.7	95.3	63.8	122.7	1449.9	296.8
R10-3	—	19.6	33.1	255.4	201.0	1225.1	654.4
R10-4	5.1	7.4	18.3	502.1	468.1	4300.4	56.3
R10-5	—	14.3	7.3	638.1	159.7	775.2	133.5
R10-6	—	97.3	72.4	375.4	1118.2	9282.8	648.3
R10-7	6.8	—	74.3	335.5	1012.8	8793.8	583.0
R10-8	1.5	—	123.1	3716.6	3140.5	18393.3	7448.6
R10-9	0.9	—	40.8	1391.6	2978.5	32881.7	5891.2
R11-1	—	229.4	66.9	3286.6	1311.5	7277.3	397.1
R11-2	—	2160.7	1377.0	3738.9	918.4	14946.7	3289.9
R11-3	18.2	2530.4	1807.6	4052.7	4360.1	18306.4	3226.8
R11-4	—	227.9	88.9	4840.5	3552.8	22116.6	1186.3
R11-5	6.7	56.3	104.9	1718.0	1064.7	16348.7	713.6
R11-6	4.8	317.9	1614.2	7635.7	3716.2	16722.5	2809.8
R11-7	1.1	—	92.0	681.6	640.7	5941.5	6261.2
R11-8	0.6	—	243.3	923.1	820.3	9661.1	1734.7
R11-9	0.4	—	117.5	113.4	156.2	2176.6	212.4
R12-1	—	460.1	272.1	4733.6	5512.9	23267.7	3880.6
R12-2	21.7	1536.2	1497.0	7292.0	5466.0	33168.4	3682.4
R12-3	—	3284.6	5746.4	11165.1	15199.8	<i>t</i>	6373.2
R12-4	12.4	77.7	98.4	3962.9	3390.1	14990.3	373.1
R12-5	19.8	69.9	475.2	4521.1	3057.8	11659.8	768.5
R12-6	—	—	1321.8	13145.8	3138.2	19674.6	3231.8
R12-7	4.3	6.8	327.8	1199.1	859.9	<i>t</i>	11603.0
R12-8	0.5	5.5	77.1	20.4	63.7	2311.6	228.3
R12-9	0.4	7.7	71.2	17.4	36.1	638.8	69.9

t:計算時間の上限; —:実行可能解なし

表 B.10 PND R 問題:個別計算時間 5

問題	閾値			アークフロー			パスフロー
	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PAFB</i> (秒)	<i>PAFS</i> (秒)	<i>PAFE</i> (秒)	<i>PPFE</i> (秒)
R13-1	—	—	18.4	72.3	146.9	167.3	20.2
R13-2	—	—	506.6	327.0	572.3	2097.0	434.2
R13-3	—	450.7	367.3	281.5	612.7	2077.3	674.4
R13-4	—	76.8	75.6	2700.8	95.2	1099.1	87.3
R13-5	—	837.3	378.5	659.6	2548.8	8107.7	1470.0
R13-6	—	798.5	1109.6	549.3	1073.7	8887.0	1659.9
R13-7	—	40.0	47.7	1724.7	754.3	10413.8	678.9
R13-8	—	80.8	120.3	477.0	3441.8	8657.3	2060.7
R13-9	—	599.3	268.9	305.2	2539.8	11785.1	1297.1
R14-1	1287.9	2438.9	576.6	11687.9	7138.8	17666.6	1882.8
R14-2	29576.4	<i>t</i>	35264.2	13622.6	6601.4	20268.1	3791.1
R14-3	5890.4	<i>t</i>	<i>t</i>	16336.8	7058.2	24499.6	3965.2
R14-4	531.0	1248.9	371.1	6529.4	1832.3	10252.1	705.9
R14-5	1705.0	9952.6	6428.1	24532.0	4978.6	18248.1	4640.6
R14-6	—	8612.3	17171.0	11018.8	11051.8	27380.5	14931.2
R14-7	—	791.5	613.5	12044.4	7468.9	27013.5	2369.3
R14-8	12.5	1646.9	2554.0	1906.2	2067.3	34232.2	1467.8
R14-9	4.2	556.2	2805.0	3334.6	2202.5	22276.5	1815.6
R15-1	<i>t</i>	<i>t</i>	13158.3	19345.6	23516.0	<i>t</i>	3661.5
R15-2	<i>t</i>	<i>t</i>	<i>t</i>	5349.2	22542.1	<i>t</i>	11224.3
R15-3	<i>t</i>	<i>t</i>	<i>t</i>	18610.6	22726.2	<i>t</i>	13392.6
R15-4	<i>t</i>	11098.4	9120.8	20044.9	27798.1	<i>t</i>	4495.1
R15-5	<i>t</i>	<i>t</i>	25689.2	19127.4	26078.9	<i>t</i>	7722.1
R15-6	—	20148.4	26191.0	12719.7	19160.2	<i>t</i>	10968.8
R15-7	188.1	331.3	861.3	7131.9	34044.1	<i>t</i>	1665.4
R15-8	—	—	2527.0	8547.9	3561.8	14553.6	2187.8
R15-9	—	—	4728.7	34192.9	3666.6	<i>t</i>	9952.7

t:計算時間の上限; —:実行可能解なし

研究業績

：主に関係する論文

：関係がある論文

国際会議論文 (査読有)

- 1) "A Model for the Optimal Number and Locations of Public Distribution Centers", Shigeru Yurimoto, Naoto Katayama, *Proceedings of the 15th International Conference on Production Research*, Vol.1, pp.677-680, 1999.
- 2) "A Location Model of Public Distribution Centers considering NOx Emissions", Shigeru Yurimoto, Naoto Katayama, *Proceedings of the 5th International Symposium on Logistics*, pp.576-581, 2000.
- 3) "The Reconciliation of Supply Chain Management with Regional Planning", Yuji Yano, Shigeru Yurimoto, Naoto Katayama, *Proceedings of the 16th International Conference on Production Research*, pp.1-10 in CD-ROM, 2001.
- 4) "The Load Planning Problem for Less-than-Truckload Motor Carriers and a Solution Approach", Naoto Katayama, Shigeru Yurimoto, *Proceedings of the 7th International Symposium on Logistics*, pp.567-572, 2002.
- 5) "Possibility of a Shift from Private to Commercial Trucks", Sunmi Yun, Naoto Katayama, Shigeru Yurimoto, *Proceedings of the 9th International Symposium on Logistics*, pp.601-608, 2004.
- 6) "A Modal Shift Model for the Logistics Network System in Japan", Sunmi Yun, Naoto Katayama, Shigeru Yurimoto, *Proceedings of the 18th International Conference on Production Research*, in CD-ROM, 2005.
- 7) "A Product-to-Plant Allocation Problem in Logistics Network Design", Naoto Katayama, Shigeru Yurimoto, *Proceedings of the 19th International Conference on Production Research*, in CD-ROM, 2007.
- 8) "A Supply Chain Network Design Problem with Concave Inventory Costs", Seyu Tei, Naoto Katayama, Shigeru Yurimoto, *Proceedings of the 20th International Conference on Production Research*, in CD-ROM, 2009.

論文(査読有)

- 1) 「道路網建設問題に対するヒューリスティック解法」, 片山 直登 , 春日井 博, 日本経営工学会誌, Vol.39, No.2, pp.69-75, 1988.
- 2) 「予算制約を持つ離散型交通ネットワークデザイン問題のラグランジュ緩和を用いた解法」, 片山 直登 , 春日井 博, 日本経営工学会誌, Vol.40, No.4, pp.237-245, 1989.
- 3) 「容量制約をもつ多品種流ネットワークデザイン問題 - 妥当不等式を用いた下界値の解法」, 片山 直登 , 春日井 博, 日本経営工学会誌, Vol.44, No.3, pp.164-175, 1993.
- 4) 「ラグランジュ緩和を用いた容量制約のないネットワーク計画問題の解法」, 片山 直登, 岩田 実, 柳下 和夫, 三原 一郎, 今澤 明男, 土木計画学研究論文集, Vol.11, pp.105-112, 1993.(第四章)
- 5) 「ラグランジュ緩和法を用いた予算制約をもつネットワークデザイン問題の解法」, 片山 直登 , 春日井 博 , 日本経営工学会誌, Vol.46, No.1, pp.21-27, 1995.
- 6) "A Model for the Optimal Number and Locations of Public Distribution Centers and its Application to the Tokyo Metropolitan Area ", Shigeru Yurimoto , Naoto Katayam , *International Journal of Industrial Engineering* , Vol.9 , No.4 , pp.363-371 , 2002.
- 7) "A Location Model of Public Distribution Centers considering NOx Emissions under Traffic Congestion ", Shigeru Yurimoto , Naoto Katayam , 日本物流学会誌 , No.10 , pp.81-89 , 2002.
- 8) 「予算制約をもつネットワークデザイン問題の貪欲解法」, 片山 直登 , 百合本 茂 , 土木計画学研究・論文集, Vol.20, No.3, pp.779-786, 2003.(第三章)
- 9) 「ロジスティクスネットワークにおける製品の工場割り当て問題」, 片山 直登, 日本物流学会誌, No.12, pp.111-118, 2004.
- 10) 「トラック輸送におけるモーダルシフトの可能性」, 尹 仙美 , 片山 直登 , 百合本 茂 , 日本物流学会誌, No.12, pp.119-126, 2004.
- 11) 「トラック輸送から鉄道・フェリー輸送へのモーダルシフトモデル」, 尹 仙美 , 片山 直登 , 百合本 茂 , 日本物流学会誌, No.13, pp.35-42, 2005.
- 12) 「施設配置を考慮したネットワークデザイン問題に対するラグランジュ緩和法」, 陳 明哲 , 片山 直登 , 毛利 裕昭 , 日本経営工学会誌, Vol.56, No.5, pp.331-338, 2005.
- 13) 「積替回数制約を持つ施設配置ネットワーク設計問題」, 陳 明哲 , 片山 直登 , 日本ロジスティクスシステム学会誌, Vol.6, No.1, pp.29-42, 2005.
- 14) 「容量制約をもつ多品種フロー輸送ネットワーク設計問題に対する容量スケールリング法」, 陳 明哲, 片山 直登, 久保 幹雄 , 日本物流学会誌, No.14, pp.85-92,

2006.(第五章)

- 15) 「区分的線形費用をもつネットワーク設計問題に対する容量スケーリング法」, 片山直登, 陳明哲, 久保幹雄, 日本ロジスティクスシステム学会誌, Vol.8, No.1, pp.3-12, 2008.(第六章)
- 16) 「ロジスティクスネットワーク設計問題に対する近似解法」, 鄭世雄, 片山直登, 日本物流学会誌, No.17, pp.165-172, 2009.
- 17) "A Capacity Scaling Procedure for the Multi-Commodity Capacitated Network Design Problem ", Naoto Katayama, Ming Zhe Chen, Mikio Kubo, *Journal of Computational and Applied Mathematics*, Vol.232, No.1, pp.90-101, 2009.(第五章)

論文 (査読無)

- 1) 「分枝限定法を用いた道路ネットワークデザイン問題の解法」, 片山直登, 春日井博, 早稲田大学理工学部研究所報告書, Vol.116, pp.1-6, 1986.
- 2) 「バス専用レーン設置問題の最適解法」, 片山直登, 春日井博, 早稲田大学理工学部研究所報告書, Vol.118, pp.12-18, 1987.
- 3) 「中央線変移問題に関する研究」, 溝口智之, 片山直登, 石井和克, 金沢工業大学研究紀要, Vol.33, pp.31-38, 1992.
- 4) 「ネットワークデザイン問題の近似解法」, 片山直登, 流通情報学部開校記念論文集, pp.171-191, 1997.(第三章)
- 5) 「利用者均衡を考慮したリバーシブル・レーン問題の近似解法」, 片山直登, 流通経済大学流通情報学部紀要, Vol.2, No.1, pp.37-49, 1997.
- 6) 「交通シミュレーションモデルの開発と事例分析」, 片山直登, 流通経済大学流通情報学部紀要, Vol.2, No.2, pp.15-29, 1998.
- 7) 「利用者均衡条件をもつ連続型ネットワークデザイン問題の近似解法」, 片山直登, 流通経済大学流通情報学部紀要, Vol.3, No.2, pp.1-14, 1999.
- 8) 「公共物流拠点配置の基礎的研究」, 片山直登, 百合本茂, 森戸晋, 矢野裕児, 苦瀬博仁, 日本物流学会誌, No.7, pp.11-13, 1999.
- 9) 「容量制約をもつネットワークデザイン問題の性質 - 双対問題, Lagrange 緩和問題, 妥当不等式 - 」, 片山直登, 流通経済大学流通情報学部紀要, Vol.4, No.2, pp.51-63, 2000.(第五章)
- 10) 「容量制約をもつ最小木問題のモデルとアルゴリズム」, 片山直登, 流通経済大学流通情報学部紀要, Vol.5, No.1, pp.21-40, 2000.
- 11) 「予算制約をもつネットワークデザイン問題の近似解法」, 片山直登, 流通経済大学流通情報学部紀要, Vol.5, No.2, pp.29-40, 2001.
- 12) 「容量制約をもつネットワークデザイン問題の貪欲解法」, 片山直登, 流通経済大学

- 流通情報学部紀要, Vol.6, No.1, pp.57-67, 2001.
- 13) 「共同輸送ネットワーク設計問題に対する Lagrange 緩和法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.6, No.2, pp.81-91, 2002.
 - 14) 「予算制約をもつネットワークデザイン問題の双対上昇法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.7, No.2, pp.17-23, 2003.
 - 15) 「分割妥当不等式を用いた容量制約をもつネットワークデザイン問題の Lagrange 緩和法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.7, No.2, pp.25-32, 2003.
 - 16) 「容量制約をもつネットワークデザイン問題の双対上昇法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.8No.2, pp.23-32, 2004.
 - 17) 「容量スケールリング法を用いた容量制約をもつ多品種ネットワークデザイン問題の近似解法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.9, No.2, pp.1-12, 2005.(第五章)
 - 18) 「利用者均衡ネットワーク設計問題に対する近似解法の調査」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.11, No.1, pp.13-30, 2006.
 - 19) 「列生成法と行生成法を用いた容量制約のないネットワークデザイン問題の近似解法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.12, No.2, pp.1-15, 2008.(第三章)
 - 20) 「強い強制制約式を用いた容量制約のないネットワークデザイン問題の Lagrange 緩和法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.13, No.1, pp.1-14, 2008.(第四章)
 - 21) 「列生成法と行生成法を用いた区分的線形関数をもつネットワークデザイン問題の近似解法」, 片山 直登, 流通経済大学流通情報学部紀要, Vol.14, No.2, pp.1-16, 2009.(第六章)

国際会議発表 (査読無)

- 1) " Lagrangian Approach for Hybrid Problem of Facility Location and Network Design ", Hiroaki Mouri, Ming Zhe Chen, Naoto Katayama, *18th International Symposium on Mathematical Programming*, pp.42, 2003.
- 2) " A Capacity Scaling Procedure for the Multi-Commodity Capacitated Network Design Problem ", Ming Zhe Chen, Naoto Katayama, Mikio Kubo, *The International Federation of Operational Research Societies*, 2005.
- 3) " A Capacity Scaling Procedure for the Capacitated Network Design Problem with Piecewise Linear Costs ", Ming Zhe Chen, Naoto Katayama, Mikio Kubo, *INFORMS International Hong Kong*, 2006.

国内学会発表

- 1) 「最適道路建設区間選定問題に関する研究」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1984.
- 2) 「道路建設区間設定問題に関する研究 - 新設路線を考慮したヒューリスティック解法」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1985.
- 3) 「非線形走行時間を考慮した最適道路建設問題に関する研究」, 片山 直登 , 春日井 博, 土木学会, 関東支部技術研究発表会講演概要集, 1986.
- 4) 「ネットワークデザインに関する研究」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1986.
- 5) 「分枝限定法を用いた道路ネットワークデザイン問題の研究」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1986.
- 6) 「バス専用レーン選定問題の基礎的研究」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1986.
- 7) 「最適道路建設問題に関する研究」, 片山 直登 , 春日井 博, 土木学会, 年次学術講演会講演概要集, 1986.
- 8) 「離散増設容量ネットワークデザイン問題の近似解法」, 片山 直登 , 春日井 博, 土木学会, 関東支部技術研究発表会講演概要集, 1987.
- 9) 「連続増設容量道路ネットワークデザイン問題」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1987.
- 10) 「需要変動を考慮した連続容量道路ネットワークデザイン問題」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1987.
- 11) 「高速デジタル回線網の最適回線数決定問題」, 片山 直登 , 関口克夫, 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1987.
- 12) 「多重容量制約付き多品種流ネットワークデザイン問題」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1988.
- 13) 「双対上昇法を用いた容量制約付き多品種流ネットワークデザイン問題の解法」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1989.
- 14) 「配送車の積載量を考慮した配送ネットワーク計画問題の一解法」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1989.
- 15) 「FixedCharge 型マルチコモディティネットワークデザイン問題」, 片山 直登 , 春日井 博, 日本オペレーションズリサーチ学会, 秋季研究発表会アブストラクト集, 1990.
- 16) 「予算制約をもつ最適ネットワークデザイン計画問題のための下界値の解法」, 片山 直登 , 春日井 博, 土木学会, 年次学術講演会講演概要集, 1991.
- 17) 「FixedCharge 型マルチコモディティネットワークデザイン問題のベンダーズカット」, 片山 直登 , 春日井 博, 日本オペレーションズリサーチ学会, 秋季研究

- 発表会アブストラクト集, 1991.
- 18) 「予算制約をもつネットワークデザイン問題」, 片山 直登 , 春日井 博, 日本経営工学会, 春季大会予稿集, 1992.
 - 19) 「中央線変移問題に関する研究」, 溝口 智之 , 片山 直登 , 石井 和克, 土木学会, 年次学術講演会講演概要集, 1992.
 - 20) 「容量制約付きネットワークデザイン問題の強い妥当不等式」, 片山 直登 , 春日井 博, 日本経営工学会, 秋季研究大会予稿集, 1992.
 - 21) 「タブサーチを用いた交通ネットワークデザイン問題の解法」, 片山 直登, 日本経営工学会, 秋季研究大会予稿集, 1993.
 - 22) 「容量制約のないネットワークデザイン問題の近似解法」, 岩田 実, 片山 直登, 中西 昌弘, 宮本 昌幸, 土木学会, 中部支部研究発表会講演概要集, pp535-536, 1994.
 - 23) 「最適通信木問題の近似解法」, 片山 直登, 日本経営工学会, 春季大会予稿集, 1994.
 - 24) 「予算制約をもつ離散型交通ネットワーク計画問題の解法」, 片山 直登, 土木学会, 年次学術講演会講演概要集, 1994.
 - 25) 「タブサーチを用いた凹費用関数をもつ施設配置フロー問題の近似解法」, 伊豆原 英樹, 片山 直登, 石井和克, 土木学会, 土木計画学研究講演集, No.17, pp.1059 - 1062, 1995.
 - 26) 「配送経路問題に関する研究」, 岩田 実, 片山 直登, 土木学会, 中部支部研究発表会講演概要集, pp457-458, 1995.
 - 27) 「選挙区区割問題の近似解法」, 片山 直登, 松本 仁志, 日本経営工学会, 春季大会予稿集, 1995.
 - 28) 「公共物流拠点配置の基礎的研究」, 片山 直登, 百合本 茂, 森戸 晋, 矢野 裕児, 苦瀬 博仁, 日本物流学会, 全国大会報告要旨集, pp.22-25, 1998.
 - 29) 「都市物流システム改善の新たな取り組み」, 高橋 洋二, 高田 邦道, 岐美宗, 苦瀬 博仁, 岩尾 詠一郎, 嶋野 崇文, 根本 敏則, 片山 直登, 百合本 茂, 土木学会, 土木計画学研究・講演集, Vol.21, No.1, pp.633-634, 1998.
 - 30) 「環境を考慮した総合物流拠点立地モデル」, 片山 直登, 百合本 茂, 日本物流学会, 全国大会報告要旨集, pp.68-71, 2000.
 - 31) 「ハブ空港配置問題の基礎的研究」, 江 浩, 片山 直登, 若林 宏明, 土木学会, 関東支部技術研究発表会講演概要集, pp.532-533, 2001.
 - 32) 「環境を考慮した総合物流拠点立地モデルの基礎的研究」, 尹 仙美, 片山 直登, 若林 宏明, 土木学会, 関東支部技術研究発表会講演概要集, pp.592-593, 2001.
 - 33) 「予算制約をもつネットワークデザイン問題の貪欲解法」, 片山 直登, 百合本 茂, 土木学会, 土木計画学研究講演集, Vol.24, No.2, pp.201-204, 2001.
 - 34) 「トラック輸送におけるモーダルシフト」, 尹 仙美, 片山 直登, 百合本 茂, 日本物流学会, 全国大会報告要旨集, pp.5-8, 2003.
 - 35) 「自家用トラックから営業用トラックへのモーダルシフトの可能性」, 尹 仙美, 片山 直登, 百合本 茂, 土木学会, 年次学術講演会講演概要集, Vol.2, pp.789-790,

2003.

- 36) 「施設配置を考慮したネットワークデザイン問題に対するラクランジュ緩和法」, 陳明哲, 片山直登, 日本経営工学会, 秋季研究大会予稿集, pp.238-239, 2003.
- 37) 「混載輸送ネットワーク設計問題に対する Lagrange 緩和法」, 片山直登, 土木学会, 土木計画学研究・講演集, Vol.28, in CD-ROM, 2003.
- 38) 「トラック輸送から鉄道・フェリーへのモーダルシフトモデル」, 尹仙美, 片山直登, 百合本茂, 日本物流学会, 全国大会報告要旨集, pp.159-162, 2004.
- 39) 「積替回数制約をもつ施設配置ネットワーク設計問題の解法」, 陳明哲, 片山直登, 日本経営工学会, 秋季研究大会予稿集, pp.124-125, 2004.
- 40) 「容量制約をもつ多品種フローネットワーク設計問題に対する容量スケールリング法」, 陳明哲, 片山直登, 久保幹雄, 日本オペレーションズリサーチ学会, 春季研究発表会アブストラクト集, pp.108-109, 2005.
- 41) 「区分的線形費用をもつ多品種流輸送ネットワーク設計問題に対する容量スケールリング法」, 陳明哲, 片山直登, 久保幹雄, 日本物流学会, 全国大会報告要旨集, pp.167-170, 2005.
- 42) 「在庫費用を考慮したロジスティクスネットワーク設計問題」, 鄭世雄, 片山直登, 日本ロジスティクスシステム学会, 全国大会予稿集, pp.117-120, 2009.
- 43) 「小売企業の配送センター立地問題 - 大連大商グループを事例に - 」, 黄大鵬, 百合本茂, 片山直登, 日本物流学会, 全国大会報告要旨集, pp.189-192, 2009.

著書・翻訳

- 1) 「基礎と実践・FORTRAN」, 広尾純, 開沼泰隆, 百合本茂, 片山直登, 技術書院, 1989.
- 2) 「新製品開発のための電撃戦略」(翻訳), ウィラード ザングウィル, 柳下和夫(監訳), 片山直登他, NTT出版, 1994.
- 3) 「ロジスティクスのOR」, 増井忠幸, 百合本茂, 片山直登, 槇書店, 1998.
- 4) 「ロジスティクスのOR ソフトウエア」, 片山直登, 槇書店, 1998.
- 5) 「経営科学OR用語大事典」(翻訳), 伊理正夫他(監訳), 片山直登他, 朝倉書店, 1999.
- 6) 「企業ロジスティクス活動の高度化が交通施設に与える影響に関する研究」, 高橋洋二, 片山直登他, 日交研シリーズ A-334, 日本交通政策研究所, 2002.
- 7) 「応用数理計画ハンドブック」, 久保幹雄他(監訳), 片山直登他, 朝倉書店, 2002.
- 8) 「ネットワーク設計問題」, 片山直登, 朝倉書店, 2008.

科学研究費・研究助成

- 1) 科学研究費補助金，一般研究 (C)，設計・生産・物流機能の総合化管理方式に関する研究，石井 和克 (代表)，片山 直登，1991.
- 2) 科学研究費補助金，一般研究 (C)，設計・生産・物流機能の統合化のための工程設計法に関する研究，石井 和克 (代表)，片山 直登，1993-1994.
- 3) 科学研究費補助金，奨励研究 (A)，パークアンドライドシステムのモデル化および解析システムの開発に関する研究，片山 直登 (代表)，1994.
- 4) 科学研究費補助金，奨励研究 (A)，総合交通システムを考慮した新交通システムのモデル化および解析システムの開発，片山 直登 (代表)，1995.
- 5) 科学研究費補助金，奨励研究 (A)，都市交通方策を考慮した統合交通システムのモデル化および解析システムの開発，片山 直登 (代表)，1996.
- 6) 日本物流学会 JILS 研究助成金，環境・社会システムを考慮した統合型物流拠点立地モデルおよび解析システムの開発，片山 直登 (代表)，1999.
- 7) 科学研究費補助金，基盤研究 (C)，統合型モーダルシフトモデルおよび解析システムの開発，片山 直登 (代表)，百合本 茂，矢野 裕児，若林 宏明，2001-2003.
- 8) 日本物流学会 JILS 研究助成金，統合型モーダルシフトモデルおよび解析システムの開発，片山 直登 (代表)，百合本 茂，2004.
- 9) 科学研究費補助金，基盤研究 (C)，環境配慮型サプライチェーンネットワーク設計モデルおよび解析システムの開発，片山 直登 (代表)，百合本 茂，河原田 秀夫，矢野 裕児，2004-2006.
- 10) 科学研究費補助金，基盤研究 (C)，環境を考慮した統合的サプライチェーンネットワーク設計問題の実用モデルと解法の開発，片山 直登 (代表)，百合本 茂，2009-2011.

謝辞

本論文の作成にあたり，御意見と御指導を頂いた方々に感謝の意を示します．

本論文をまとめるにあたり有益なる御指導を賜り，かつ貴重な時間を割いて論文審査をして頂いた流通経済大学 百合本茂教授，流通経済大学 増田悦夫教授，東京海洋大学 久保幹雄教授に御礼を申し上げます．

早稲田大学在籍時に，研究のきっかけを与えて頂き，研究を指導して頂いた元早稲田大学 故春日井博教授に感謝致します．共同研究と研究指導を通じて，多くの成果を分かち合うことができた陳明哲博士，尹仙美博士に深く感謝致します．また，貴重なデータを提供して頂いたモントリオール大学 Crainic 教授に御礼を申し上げます．早稲田在籍中からの研究活動において，また本論文に関係する研究を行うにあたって，多くの有意義な助言と考えを提供していただいた東京海洋大学 久保幹雄教授に再度感謝致します．