

実行不可能解を活用する  
進化型制約付き多目的最適化

宮川 みなみ

電気通信大学 大学院情報理工学研究科  
博士(工学)の学位申請論文

2016年3月



実行不可能解を活用する  
進化型制約付き多目的最適化

論文審査委員会

主査	高玉	圭樹	教授
委員	西野	哲朗	教授
委員	高橋	治久	教授
委員	庄野	逸	教授
委員	高橋	裕樹	准教授
委員	佐藤	寛之	助教

Copyright (C) 2016 Minami MIYAKAWA All Rights Reserved.

# Abstract

For solving optimization problems with multiple objectives and constraints, this thesis proposes a novel evolutionary algorithm framework utilizing infeasible solutions not satisfying the constraints but having better objective function values than feasible ones. First, the base algorithm using the two-stage non-dominated sorting and the directed mating (TNSDM) is proposed. To rank the solutions in the population, the two-stage non-dominated sorting classifies the population into several sets of solutions based on the constraint violation values, then reclassifies each of them based on the objective values. To generate offspring, the directed mating selects a feasible solution as the primary parent, picks infeasible solutions having better objective values than the primary parent, and selects the secondary parent from them based on the rank of the solutions obtained by the two-stage non-dominated sorting. Thus, the directed mating in TNSDM utilizes useful infeasible solutions having better objective values than feasible ones as parents.

First, this thesis shows that the directed mating in TNSDM contributes to improving the search performance of evolutionary constrained multi-objective optimization by utilizing useful infeasible solution having better objective values than feasible ones. Second, to overcome a problem that the effectiveness of the directed mating depends on characteristics of problems, TNSDM-CS controlling selection area of useful infeasible solutions is proposed. For problems with less infeasible solutions, TNSDM-CS expands selection areas of useful infeasible solutions and encourages to execute the directed mating. For problems with much infeasible solutions, TNSDM-CS contracts selection areas and strengthens the directivity of the solution search. Third, to tackle a problem that useful infeasible solutions are discarded from the population every generation since they are infeasible even if they have good objective values, TNSDM-A archiving useful infeasible solutions in the population and repeatedly utilizing them as parents is proposed. Fourth, TNSDM-CSA combining TNSDM-CS and -A is proposed. Contracted selection areas have a positive effect to improve the search performance since the directivity of the solution search is strengthened, but have a negative effect that the number of directed mating executions is decreased since several primary parents cannot pick

secondary parent candidates. When the archive of useful infeasible solutions is combined, almost all primary parents can pick several parent candidates even if selection areas are contracted. Fifth, TNSDM-CG controlling the number of genes (variables) inherited from useful infeasible solutions is proposed. When the number of genes inherited from useful infeasible solutions is decreased, generated offspring tend to be feasible but useful genetic information is decreased. On the other hand, the number of genes inherited from useful infeasible solutions is increased, useful genetic information copied from infeasible solutions is increased but generated offspring tend to be infeasible. TNSDM-CG controls the balance between them and improves the effectiveness of the directed mating. Finally, the search performance of TNSDM-CSACG combining TNSDM-CS, -A and -CG is verified.

To investigate effects of the proposed algorithms,  $m$ -objective  $k$ -knapsack problem as a discrete test problem, SRN, OSY, TNK, mCDTLZ as continuous test problems, and a conceptual design optimization problem of three stage hybrid rocket as a real-world problem are employed. As conventional algorithms, representative CNSGA-II and RTS are used in this work. Experimental results show that all algorithms proposed in this thesis achieve higher search performance than the conventional algorithms. The details are as follows. First, experimental results clarify that the basic TNSDM achieves better search performance than the conventional algorithms. It reveals that the search performance of evolutionary constrained multi-objective optimization is improved by the directed mating utilizing useful infeasible solutions having better objective values than feasible ones. Second, results show that TNSDM-CS further improves the effectiveness of the directed mating by controlling selection areas of useful infeasible parents. Expanded selection areas have an effect to increase the number of the directed mating executions, contracted selection areas have an effect to strengthen the directivity of the solution search, and each effect contribute to improving the search performance. Third, experimental results clarify that TNSDM-A also improves the search performance by maintaining useful infeasible solutions in the population and repeatedly utilizing useful infeasible solutions which are discarded from the population every generation in the basic TNSDM. Fourth, results reveal that TNSDM-CSA achieves higher search performance than TNSDM-CS and TNSDM-A by increasing the number of the directed mating executions with the archive and further contracting selection areas and strengthening the directivity of the solution search. Fifth, the experimental results show that TNSDM-CG also enhances the effectiveness of the directed mating by controlling the feasibility of generated offspring and the amount of genes involving useful genetic information. Finally, the thesis shows that TNSDM-CSACG combining TNSDM-CS, -A, and -CG achieves the highest search performance among TNSDM series proposed in this work.

# 概要

本研究では，進化計算による制約付き多目的最適化において，従来法では単に排除される制約条件を満たさない実行不可能解を解探索に有効活用する方法について探求し，その有効性を計算機実験によって検証することを目的とする．この目的を達成するために，まず，本研究の基礎となる二段階の非支配ソートと指向性交配を用いる TNSDM (Two-stage Non-dominated Sorting and Directed Mating) アルゴリズムを提案する．二段階の非支配ソートでは，制約違反量に基づいて解集団をいくつかのグループに分類した後，各グループを目的関数値に基づいて再分類することによって解をランキングする．指向性交配では，実行可能解を第一の親とし，それより目的関数値が良い実行不可能解集合を選出する．その中から，二段階の非支配ソートで求めた解のランキングに従って第二の親を選択し，子を生成する．制約条件を満たさない実行不可能解の中には，解集団中の実行可能解より良い目的関数値 (評価値) を示す場合があるため，それを解探索の手掛かりに活用する．

まず，(1) 基礎となる TNSDM の指向性交配において，実行可能解より高い目的関数値を持つ解を解探索に活用することによって最適化性能が改善されることを示す．次に，(2) 指向性交配による解探索効果が問題の性質に依存することを解決するために，有用な実行不可能解の選出領域を制御する TNSDM-CS (TNSDM with Controlling Selection area) を提案する．これにより，解探索中に生じる実行不可能解が少ない問題では，選出領域を拡大して指向性交配を実行しやすく，多い問題では縮小して良い親を選べるようになる．また，(3) 指向性交配で活用する実行可能解は，良い目的関数値を持つが実行不可能であるがゆえ，世代ごとに解集団から消失される．この問題を解決するため，有用な実行不可能解を解集団中にアーカイブし，解探索の手掛かりとして繰り返し活用する TNSDM-A (TNSDM with Archive) を提案する．さらに，(4) TNSDM-CS と TNSDM-A を組み合わせた TNSDM-CSA (TNSDM with Controlling Selection area and Archive) を提案する．TNSDM-CS において，有用な実行不可能解の選出領域を縮小すると，解探索の指向性が高まるため解探索性能が改善される利点があるが，同時に，選出領域に解が存在せずに指向性交配を実行できないケースが生じる欠点がある．この問題に対して，有用な実行不可能解のアーカイブを導入すると，選出領域を縮小しても選出領域に解が得られやすくなり，選出領域を縮小する欠点を補える．また，(5) 指向性交配で選択する有用な実行不可能解

から子に複写される遺伝子(変数)の量を操作する TNSDM-CG (TNSDM with Controlling crossed Genes) を提案する。実行不可能解から子に複写される遺伝子量を減少させると、子は実行可能解になりやすいが有用な遺伝子情報を得にくい。逆に増加させると、子は実行不可能解になりやすいが有用な遺伝子情報を得やすくなる。このバランスを操作することによって、指向性交配の解探索効果を高める。最後に、(6)TNSDM-CS と TNSDM-A, TNSDM-CG それぞれの指向性交配の効果を高める手法を組み合わせた TNSDM-CSACG (TNSDM with Controlling Selection area, Archive and Controlling crossed Genes) を提案し、その効果を検証する。

提案法の効果を検証するため、離散問題の  $m$  目的  $k$  ナップザック問題、連続問題の SRN, OSY, TNK, mCDTLZ, 実問題の 3 段式ハイブリッドロケットの概念設計最適化問題を用いる。また、比較アルゴリズムとして、代表的な CNSGA-II と RTS アルゴリズムを用いる。実験の結果、本研究で提案するいずれのアルゴリズムも、従来法より高い解探索性能を示すことが明らかになった。その詳細について順に述べると、まず、(1) TNSDM が従来法より高い解探索性能を示すことがわかった。これより、目的関数値の高い実行不可能解を活用する指向性交配によって、制約条件を有する多目的最適化問題における進化計算の解探索性能が改善されることが明らかになった。次に、(2) TNSDM-CS において有用な実行不可能解の選出領域を制御することによって、指向性交配による解探索効果がさらに高まることが明らかになった。選出領域が拡大されると指向性交配の実行回数が増加する効果があり、縮小されると解探索の指向性が高まる効果があるため、解探索性能が改善される。また、(3) TNSDM-A において有用な実行不可能解を解集団中にアーカイブすることによっても、指向性交配による解探索効果がさらに高まることが明らかになった。これは、TNSDM では世代ごとに消失してしまう有用な実行不可能解を繰り返し指向性交配に活用できるためである。さらに、(4) TNSDM-CSA において有用な実行不可能解の選出領域制御法にアーカイブ法を組み合わせると、指向性交配を実行できないケースが減るため、より選出領域を縮小して解探索の指向性を高めた場合に、TNSDM-CS と TNSDM-A より高い解探索性能を達成できることが明らかになった。また、(5) TNSDM-CG において有用な実行不可能解から子へ複写される遺伝子(変数)量を操作することによって、子の実行可能解へのなりやすさと、子に複写される解探索に有用な遺伝子情報の量を操作できるようになり、指向性交配の効果が高まることが明らかになった。最後に、(6)TNSDM-CSACG において、有用な実行不可能解の選出領域制御法とアーカイブ法、交叉量操作法を組み合わせると、上記(1)~(5)のアルゴリズムと比べ、最も高い解探索性能を示すことが明らかになった。(4)において示される有用な実行不可能解の選出領域制御法にアーカイブ法の組み合わせによる効果に加え、交叉量操作により親として選択された実行不可能解から子に複写する遺伝子量を制御することによって解探索性能が高まる。



# 目次

第 1 章	序論	1
1.1	研究背景	1
1.2	研究目的と方法	2
1.3	本論文の構成	3
第 2 章	制約なし多目的最適化問題と進化計算による解法	6
2.1	制約なし多目的最適化問題	6
2.2	進化計算法	7
2.3	NSGA-II	12
第 3 章	制約付き多目的最適化問題と進化計算による解法	14
3.1	制約付き多目的最適化問題	14
3.2	制約付き問題に対する進化計算法	15
3.3	本研究の位置づけ	20
第 4 章	テスト問題と評価尺度	24
4.1	テスト問題の扱い	24
4.2	離散テスト問題	24
4.3	連続テスト問題	26
4.4	評価尺度	29
第 5 章	二段階の非支配ソートと指向性交配を用いる MOEA	35
5.1	TNSDM アルゴリズム	35
5.2	実験内容	38
5.3	実験結果と考察	40
5.4	3 段階ハイブリッドロケットの概念設計最適化における解探索性能	47
5.5	計算時間	49
5.6	動的計画法との比較	50
5.7	真の最適解との比較	51

5.8	まとめ . . . . .	52
<b>第 6 章</b>	<b>TNSDM における有用な実行不可能解の選出領域制御</b>	<b>53</b>
6.1	指向性交配の問題 1 : 交配する候補解集合の選出領域 . . . . .	53
6.2	TNSDM-CS : 有用な実行不可能解の選出領域を制御する TNSDM . . . . .	53
6.3	実験内容 . . . . .	57
6.4	実験結果と考察 . . . . .	59
6.5	解の選出領域の適応制御 . . . . .	65
6.6	まとめ . . . . .	69
<b>第 7 章</b>	<b>TNSDM における有用な不可能解のアーカイブ法</b>	<b>70</b>
7.1	指向性交配の問題 2 : 有用な実行不可能解の消失 . . . . .	70
7.2	TNSDM-A : 有用な実行不可能解をアーカイブする TNSDM . . . . .	70
7.3	実験内容 . . . . .	72
7.4	実験結果と考察 . . . . .	75
7.5	まとめ . . . . .	80
<b>第 8 章</b>	<b>有用な実行不可能解の選出領域制御法とアーカイブ法の組み合わせ</b>	<b>81</b>
8.1	選出領域制御法とアーカイブ法を組み合わせる意義 . . . . .	81
8.2	TNSDM-CSA : 選出領域制御法とアーカイブ法を用いる TNSDM . . . . .	81
8.3	実験内容 . . . . .	82
8.4	実験結果と考察 . . . . .	84
8.5	まとめ . . . . .	88
<b>第 9 章</b>	<b>TNSDM における交叉量操作法</b>	<b>89</b>
9.1	指向性交配の問題 3 : 遺伝子変異量 . . . . .	89
9.2	TNSDM-CG : 指向性交配のための交叉量操作法を用いる TNSDM . . . . .	90
9.3	実験内容 . . . . .	94
9.4	実験結果と考察 . . . . .	96
9.5	まとめ . . . . .	99
<b>第 10 章</b>	<b>有用な実行不可能解の選出領域制御法とアーカイブ法, 交叉量操作法の 組み合わせ</b>	<b>101</b>
10.1	TNSDM-CSACG : 選出領域制御法とアーカイブ法, 交叉量操作法を用い る TNSDM . . . . .	101
10.2	実験内容 . . . . .	102
10.3	実験結果と考察 . . . . .	104

10.4	まとめ . . . . .	107
第 11 章	結論	108
11.1	得られた知見 . . . . .	108
11.2	今後の課題と展望 . . . . .	111
謝辞		114
参考文献		115
関連発表		120

# 目次

1.1	本研究の全体像 . . . . .	4
2.1	支配の概念図 . . . . .	7
2.2	フロント分類の概念図 . . . . .	8
2.3	一点交叉の例 . . . . .	10
2.4	複数点交叉 (二点交叉) の例 . . . . .	10
2.5	一様交叉の例 . . . . .	11
2.6	ビット反転突然変異の例 . . . . .	12
2.7	NSGA-II のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団) . . . . .	13
3.1	CNSGA-II のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団) . . . . .	17
3.2	RTS-O のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{Q}$ : 子集団) . . . . .	18
3.3	RTS-C のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{Q}$ : 子集団) . . . . .	19
3.4	MOEA/D の概念図 . . . . .	20
3.5	親の選択領域と解の探索方向 . . . . .	22
4.1	$m$ 目的 $k$ ナップザック問題 ( $mk$ -KP) . . . . .	25
4.2	$mk$ -KP における制約数 $k$ , 実行可能率 $\phi$ に対する実行可能解数の変化 . . . . .	26
4.3	mCDTLZ 問題の目的空間 ( $m = 2$ 目的) . . . . .	28
4.4	獲得した解集合とその評価 . . . . .	29
4.5	Hypervolume の概念図 . . . . .	30
4.6	弱優位の例 . . . . .	31
4.7	強優位の例 . . . . .	31
4.8	完全優位の例 . . . . .	31
4.9	Maximum Spread の概念図 . . . . .	32
4.10	Norm の概念図 . . . . .	33
5.1	TNSDM のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団) . . . . .	36
5.2	親 $p_a$ の探索方向を考慮した指向性交配 (最大化問題の例) . . . . .	39

5.3	連続問題における $HV$ の世代推移 (総合評価) . . . . .	40
5.4	連続問題における $MS$ の世代推移 (多様性評価) . . . . .	41
5.5	連続問題における $C$ メトリック比較の世代推移 (収束性評価) . . . . .	41
5.6	$mk$ -KP における最終世代の $HV$ (総合評価) . . . . .	42
5.7	$mk$ -KP における最終世代の $MS$ (多様性評価) . . . . .	43
5.8	$mk$ -KP における $C$ メトリックによる比較 (収束性評価) . . . . .	43
5.9	指向性交配によって生成された子の目的関数空間における分布 ( $mk$ -KP) . . . . .	44
5.10	二段階ソートによる解のランキングの効果 ( $mk$ -KP) . . . . .	46
5.11	3 段式ハイブリッドロケットの概念設計問題における $HV$ の世代推移 . . . . .	47
5.12	3 段式ハイブリッドロケットの概念設計問題において獲得した解集合 . . . . .	47
5.13	CNSGA-II の計算コスト (ビッグ・オー) . . . . .	48
5.14	TNSDM の計算コスト (ビッグ・オー) . . . . .	48
5.15	$mk$ -KP における計算時間の比較 . . . . .	49
5.16	$mk$ -KP において動的計画法と TNSDM により獲得した解集合 . . . . .	50
5.17	$mk$ -KP における真の最適解集合と TNSDM が獲得した解集合 . . . . .	52
6.1	TNSDM-CS のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団) . . . . .	54
6.2	CDAS による目的関数値の変換 . . . . .	57
6.3	従来の選出領域 ( $S = 0.5$ ) . . . . .	58
6.4	拡大した選出領域 ( $S < 0.5$ ) . . . . .	58
6.5	縮小した選出領域 ( $S > 0.5$ ) . . . . .	58
6.6	$mk$ -KP における指向性交配の実行割合 . . . . .	59
6.7	$mk$ -KP における最終世代の $HV$ . . . . .	60
6.8	$mk$ -KP における最終世代の $MS$ . . . . .	62
6.9	$mk$ -KP における最終世代の $Norm$ . . . . .	62
6.10	mCDTLZ 問題における指向性交配の実行割合 . . . . .	63
6.11	mCDTLZ 問題における最終世代の $HV$ . . . . .	63
6.12	$mk$ -KP における解探索性能 ( $HV$ ) と指向性交配の実行回数の割合 . . . . .	65
6.13	mCDTLZ 問題における解探索性能 ( $HV$ ) と指向性交配の実行回数の割合 . . . . .	65
6.14	解の選出領域の適応制御の概念図 . . . . .	66
6.15	選出領域の適応制御法による最終世代の $HV$ (総合評価) . . . . .	67
6.16	適応制御した $S$ の推移 ( $m = 2$ 目的問題) . . . . .	69
7.1	TNSDM-A のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団, $\mathcal{A}$ : アー カイク集団) . . . . .	71
7.2	$mk$ -KP における最終世代の $HV$ . . . . .	74

7.3	$mk$ -KP における最終世代の $MS$ . . . . .	74
7.4	$mk$ -KP における $C$ メトリックによる比較 . . . . .	76
7.5	mCDTLZ における最終世代の $HV$ . . . . .	77
7.6	mCDTLZ における最終世代の $MS$ . . . . .	77
7.7	mCDTLZ における $C$ メトリックによる比較 . . . . .	77
7.8	$mk$ -KP におけるアーカイブ集団サイズ $ \mathcal{A} $ の世代推移 ( $m = 2$ 目的) . . . . .	78
7.9	mCDTLZ におけるアーカイブ集団サイズ $ \mathcal{A} $ の世代推移 . . . . .	78
7.10	解集団サイズの変化に対する $HV$ . . . . .	79
7.11	取り扱う解集団サイズ . . . . .	79
8.1	TNSDM-CSA のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団, $\mathcal{A}$ : アーカイブ集団)) . . . . .	82
8.2	指向性交配の実行割合 ( $m = 2$ 目的問題) . . . . .	85
8.3	指向性交配の実行割合 ( $m = 4$ 目的問題) . . . . .	85
8.4	指向性交配の実行割合 ( $m = 6$ 目的問題) . . . . .	85
8.5	最終世代の $HV$ ( $m = 2$ 目的問題) . . . . .	86
8.6	最終世代の $HV$ ( $m = 4$ 目的問題) . . . . .	86
8.7	最終世代の $HV$ ( $m = 6$ 目的問題) . . . . .	86
9.1	指向性交配と従来の交配法による子の実行可能率 . . . . .	90
9.2	指向性交配と従来の交配法による $HV$ の結果 . . . . .	90
9.3	TNSDM-CG のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団) . . . . .	91
9.4	一様交叉を基にした交叉量操作 . . . . .	93
9.5	SBX を基にした交叉量操作 . . . . .	94
9.6	$mk$ -KP における生成された子の実行可能率 . . . . .	95
9.7	$mk$ -KP における最終世代の $HV$ . . . . .	96
9.8	TNK における最終世代の $HV$ と子の実行可能率 . . . . .	97
9.9	mCDTLZ における最終世代の $HV$ と子の実行可能率 . . . . .	99
10.1	TNSDM-CSACG のブロック図 ( $\mathcal{R}$ : 全解集団, $\mathcal{P}$ : 親集団, $\mathcal{Q}$ : 子集団, $\mathcal{A}$ : アーカイブ集団)) . . . . .	102
10.2	最終世代の $HV$ ( $m = 2$ 目的問題) . . . . .	105
10.3	$mk$ -KP における最終世代の $HV$ ( $m = 4$ 目的問題) . . . . .	105

# 表目次

3.1	制約付き最適化問題の進化計算による解法の第一分類 . . . . .	21
3.2	分類 <b>B</b> の第 2 フェーズにおける細分類 . . . . .	21
4.1	テスト問題のパラメータ . . . . .	27
5.1	擬似コード内で使用される記号 (TNSDM) . . . . .	38
5.2	獲得した真の最適解の個数 . . . . .	51
6.1	擬似コード内で使用される記号 (TNSDM-CS) . . . . .	56
6.2	TNSDM-CS における最適な $S^*$ . . . . .	68
7.1	擬似コード内で使用される記号 (TNSDM-A) . . . . .	73
8.1	擬似コード内で使用される記号 (TNSDM-CSA) . . . . .	84
9.1	擬似コード内で使用される記号 (TNSDM-CG) . . . . .	93
10.1	擬似コード内で使用される記号 (TNSDM-CSACG) . . . . .	104





# 第 1 章

## 序論

### 1.1 研究背景

実世界の最適化問題の多くは，製品の「性能」は最大化したいが「価格」は最小化したいというように，複数の目的の同時最適化を求められる多目的最適化問題 (Multi-objective Optimization Problem : MOP) になる．製品の性能を高めれば，価格は高くせざるを得ず，製品の価格を低くすれば，性能は落とさざるを得ない．理想の性能と価格の間にはトレードオフの関係があり，最高性能と最低価格を同時に満足する製品は，現実には存在しない．このように，MOP のゴールは，目的間の最適なトレードオフ関係を示すパレートフロントを近似する解集合を獲得することである [1, 2]．

MOP を解く古典的なアプローチとしては，解の持つ複数の目的関数値ベクトルに重み付してスカラー化することや，選好によって決定する目的関数の目標値ベクトルとの距離を用いることで多目的最適化問題を単一目的最適化問題に変換し，最適化法として，線形計画法や動的計画法，勾配法，分枝限定法などを適用する方法がとられてきた [1]．しかし，これらの方法では，一度の最適化で一つの解しか獲得できず，パレートフロントを近似するためには，重みや目標値ベクトルを徐々に変更しながら最適化を繰り返すほかない．また，それぞれの最適化手法は解ける問題の線形性や微分可能性などの特徴に制限があり，単一目的化された問題によっては適用できない場合がある．それに対して，生物の遺伝と進化を基礎とする進化計算は，解集団を用いた多点探索法であるため，パレートフロントを近似する解集合を解集団から一括獲得できる点で，多目的最適化の有用な解法手段である．また，問題の特徴を考慮する必要がなく，さらに目的関数がブラックボックスであっても適用できるところに特徴がある．遺伝的アルゴリズム (Genetic Algorithm : GA) [3, 4]，遺伝的プログラミング (Genetic Programming : GP) [5]，進化戦略 (Evolutionary Strategy : ES) [6] が代表的だが，近年では多点探索という共通点から，差分進化 (Differential Evolution : DE) [7]，粒子群最適化 (Particle Swarm Optimization : PSO) [8]，蟻コロニー最適化 (Ant Colony Optimization : ACO) [9] なども進化計算とともに議論

される。進化計算は、問題に対する事前知識を必要とせずに適用できる利点から、産業界の新しい設計手段として注目を集め [10]、新幹線 N700 系の先頭車の形状最適化 [11] や、ジェット旅客機 MRJ における翼の形状設計 [12] などに利用されはじめている。

しかし、実世界の最適化問題で考慮すべきなのは複数の目的だけではなく、「開発期間は 1 ヶ月以内」や「故障率は 0.01% 以下」といった複数の制約条件も同時に考慮しなければならない。この問題を、複数制約条件付き多目的最適化問題 (Constrained Multi-objective Optimization Problem : CMOP) という [13]。CMOP においては、制約条件を満たさない解を実行不可能解、制約条件を満たす解を実行可能解という。CMOP のゴールは、制約条件を満たす実行可能解集合の中から、目的間の最適なトレードオフ関係であるパレートフロントを近似する解集合を獲得することになる。目的が複数になるだけでも最適化が困難になることに加えて、さらに複数の厳しい制約条件が課されると、可能解すら容易に獲得できない場合もあり、最適化の困難さが増す。今後、産業界における進化計算の適用可能範囲を拡大させるためには、複数の目的と制約条件を持つ CMOP を効果的に解く方法論の構築が求められる。

CMOP の進化計算による解法では、実行不可能解の扱い方が最適化性能に大きな影響を与える。代表的手法である CNSGA-II (Constrained fast Non-dominated Sorting Genetic Algorithm) [14] による CMOP の解探索過程は、大きく 2 つに分かれる。第 1 フェーズでは、解集団中の実行不可能解を実行可能解へ進化させることを目的とする。第 2 フェーズでは、得られた実行可能解の目的関数値 (評価値) を改善するように進化させてパレートフロントの近似精度を高めることを目的とする。本研究が注目するのは、第 2 フェーズである。従来法の多くは、解集団中に実行可能解が見つかり、実行可能解の情報を主として解探索するように設計されている。解探索中には、制約条件を満たさない実行不可能解も生成されるが、それらは単に捨てられる。特に工学設計の最適化では、風洞シミュレーションなど、解の評価に時間を要するため、生成された解が実行不可能解であったとしても、それをそのまま捨てることは、計算時間を無駄にすることに他ならない。これまで単に捨てていた実行不可能解から、解探索の手掛かりとなる情報を適切に引き出して解探索に役立てることができれば、評価シミュレーションの時間が無駄にならないばかりか最適化を促進する効果が得られる可能性がある。

## 1.2 研究目的と方法

本研究では、制約付き多目的最適化問題における進化計算の解探索性能を改善するために、これまで解集団から排除されてきた制約条件を満たさない実行不可能解を解探索に有効活用する方法を構築し、その有効性を明らかにすることを目的とする。実行不可能解の活用によって、これまでより短い時間で最適化するだけでなく、実行可能解のみに基づ

く解探索では得ることが出来ない良い解の獲得を目指す。

まず、二段階の非支配ソートと指向性交配を用いる TNSDM (Two-stage Non-dominated Sorting and Directed Mating) アルゴリズムを提案し、本研究の基礎アルゴリズムとする。二段階の非支配ソートでは、制約違反量に基づいて解集団をいくつかの解集合に分類した後、各解集合を目的関数値に基づいて再分類することによって解をランキングする。指向性交配では、実行可能解を第一の親とし、それより目的関数値が良い実行不可能解集合を選出する。その中から、二段階の非支配ソートで求めた解のランキングに従って第二の親を選択し、子を生成する。制約条件を満たさない実行不可能解の中には、解集団中の実行可能解より良い目的関数値 (評価値) を示すものがあるため、それを解探索の手掛かりとして活用する。この指向性交配を、本研究の焦点である実行不可能解を活用する手段の基礎とし、以降は指向性交配による解探索効果を高める方法について検討する。まず、実行可能解と交配する実行不可能解の目的空間における選出領域の制御法について検討する。次に、実行可能解より目的関数値が良い実行不可能解を解集団中にアーカイブして、繰り返し解探索の手掛かりとして活用する方法について検討する。最後に、有用な実行不可能解の親から子へ複製される遺伝子 (変数) 量を操作する交叉法について検討する。

本研究で提案する方法の効果は、複数のテスト問題を用いて検証する。まず、離散問題として  $m$  目的  $k$  ナップザック問題 ( $mk$ -KP) [15] を用いる。この問題は、組み合わせ最適化問題の代表的なモデルであり、実世界問題の投資案件選択問題、製造オーダー選択問題、パッキング対象製品選択問題などが、ナップザック問題に帰着する。また、 $m$  目的  $k$  ナップザック問題は、目的数  $m$ 、制約 (ナップザック) 数  $k$ 、実行可能率  $\phi$  をパラメータによって指定でき、問題の設定に自由度がある。また、近年盛んに検討されている工学設計最適化で用いられる解を実数値で表現する連続問題として、本研究では、SRN [16]、TNK [17]、OSY [18]、mCDTLZ [19]、さらに実問題の 3 段式ハイブリッドロケットの概念設計問題 [20] を用いる。次に、提案法の解探索性能を比較する従来法として、本研究では、最も代表的な CNSGA-II [14] と RTS [21] を取り上げる。

### 1.3 本論文の構成

まず、2~4 章において本論文の展開に必要な基礎的事項について述べ、5 章以降において本論文で提案する方法について述べる。本研究で提案する方法の全体像を図 1.1 に示す。本論文の構成の詳細は、以下のとおりである。

第 2 章では、制約条件を持たない多目的最適化問題と、その進化計算による解法について説明する。まず、多目的最適化問題と求めるべき解を数学的に定義する。次に、進化計算の処理手順と、一般的に用いられる適応度の与え方、親選択法、交叉法、突然変異法について説明する。最後に、制約条件を持たない多目的最適化のための代表的な進化計算法

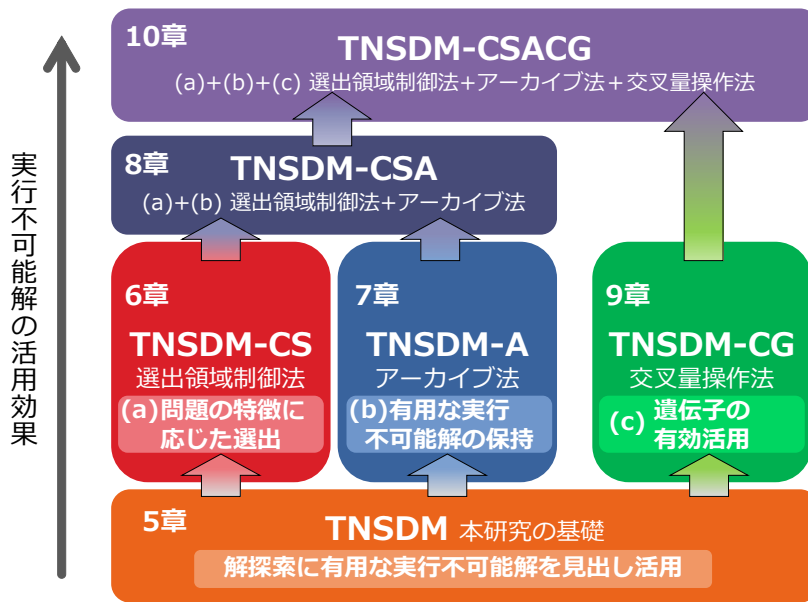


図 1.1: 本研究の全体像

として NSGA-II (fast Non-dominated Sorting Genetic Algorithm) を取り上げ、そのアルゴリズムを説明する。

第 3 章では、制約条件を持つ多目的最適化問題と、進化計算における制約条件の取り扱い方について説明する。まず、制約条件を持つ多目的最適化問題と求めるべき解を数学的に定義する。次に、進化計算における従来の制約条件の取り扱い方について説明する。最後に、従来法の特徴を整理した上で、本研究の位置づけについて述べる。

第 4 章では、本研究で用いるテスト問題と、最適化の結果として得られる解集合の評価尺度について説明する。テスト問題としては、離散問題の *mk-KP*、連続問題の *SRN*、*OSY*、*TNK*、*mCDTLZ*、実問題の 3 段式ハイブリッドロケットの概念設計問題について説明する。次に、パレートフロントを精緻に近似するために解集合が有すべき収束性と多様性、解の数、分布の一樣さの概念について述べ、それらを定量的に評価する複数の尺度について説明する。

第 5 章では、本研究の基礎アルゴリズムである二段階の非支配ソートと指向性交配を用いる TNSDM アルゴリズムを提案する。TNSDM の設計思想、詳細な処理手順について説明をした後、従来法である CNSGA-II, RTS との性能比較実験を行い、その有効性を明らかにする。また、TNSDM によって生成される解 (子) の特徴について詳細に検証する。さらに、*mk-KP* における TNSDM と CNAGA-II、また、動的計画法との計算時間の比較や、TNSDM が獲得する解と真の最適解との比較を行う。

第 6 章では、TNSDM における指向性交配の解探索効果を高めるため、有用な実行不可能解の目的空間における選出領域を拡大・縮小することによって、制約条件の厳しさの異

なる様々な問題に対して、指向性交配を実行しやすくしたり、解探索の指向性を高めたりする TNSDM-CS (TNSDM with Controlling Selection area) を提案する。選出領域を制御する方法と期待される効果について説明した後、性能比較実験を行い、提案する選出領域制御法によって、指向性交配による解探索効果が向上することを明らかにする。さらに、選出領域を適応的に制御する方法を提案し、その効果を明らかにする。

第 7 章では、有用な実行不可能解を解集団中にアーカイブする方法を提案する。指向性交配で活用する実行不可能解は、目的関数値 (評価値) は良いものの、実行不可能であるがゆえ、世代ごとに解集団から淘汰されてしまう。有用な実行不可能解を繰り返し解探索の手掛かりとして活用するため、それらをアーカイブする TNSDM-A (TNSDM with Archive) を提案する。TNSDM と解探索性能を比較し、有用な実行不可能解をアーカイブして繰り返し指向性交配に活用する効果を明らかにする。

第 8 章では、6 章で提案する有用な実行不可能解の選出領域制御法と、7 章で提案するアーカイブ法を組み合わせる TNSDM-CSA (TNSDM with Controlling Selection area and Archive) の効果について検証する。二つの方法を組み合わせる意義とそのアルゴリズムについて説明した後、実験を通して、二つの方法を組み合わせた場合の相乗効果を明らかにする。

第 9 章では、有用な実行不可能解から子に複写される遺伝子 (変数) 量を操作する TNSDM-CG (TNSDM with Controlling crossed Genes) を提案する。実行不可能解から子に複写される遺伝子 (変数) 量を減少させると、子は実行可能解になりやすいが、目的関数値を高める有用な遺伝子情報が得られにくくなる。逆に、実行不可能解から子に複写される遺伝子 (変数) 量を増加させると、子は実行不可能解になりやすいが、目的関数値を高める有用な遺伝子情報を得やすくなる。交叉する遺伝子量を操作することによって、指向性交配による解探索効果がさらに改善することを明らかにする。

第 10 章では、6, 7, 9 章において提案する TNSDM における有用な実行不可能解の選出領域制御法とアーカイブ法、交叉量操作法をすべて組み合わせる TNSDM-CSACG (TNSDM with Controlling Selection area, Archive and Controlling crossed Genes) の効果を検証する。そのアルゴリズムについて説明した後、実験を通じて三つの手法を組み合わせたときの効果を明らかにする。

最後に、第 11 章で本論文のまとめを述べ、今後の課題と展望を示す。

## 第 2 章

# 制約なし多目的最適化問題と進化計算による解法

### 2.1 制約なし多目的最適化問題

制約なし多目的最適化問題 (Multi-objective Optimization Problems : MOP) は,  $m$  種類の目的関数  $f_i$  ( $i = 1, 2, \dots, m$ ) を最大化 (あるいは最小化) する解  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  を見出す問題である. MOP は, 以下のように定義される.

$$\text{Maximize/Minimize } f_i(\mathbf{x}) \quad (i = 1, 2, \dots, m) \quad (2.1)$$

また, 最適化する目的関数がひとつの場合, 単一目的最適化という. 単一目的最適化では, 真の最適値を一意に定めることができる. 一方, 目的関数が複数の場合, 一般的に  $m$  種類の目的関数の間にはトレードオフの関係があるため, すべての目的関数を同時に最適化する単一の解は存在しない. そこでパレート支配という概念を用いて解の優劣を決定する. 最大化問題において, 解  $\mathbf{x}$ ,  $\mathbf{y}$  が次式を満たすとき,  $\mathbf{x}$  は  $\mathbf{y}$  を目的関数値において支配する ( $\mathbf{x} \succ_f \mathbf{y}$ ) という.

$$\forall i : f_i(\mathbf{x}) \geq f_i(\mathbf{y}) \quad \wedge \quad \exists i : f_i(\mathbf{x}) > f_i(\mathbf{y}) \quad (i = 1, 2, \dots, m) \quad (2.2)$$

なお, 最小化問題の場合は, 式 (2.2) の不等号を逆にする. ここで, 図 2.1 に 2 目的最大化問題における支配関係の概念図を示す. 図中の丸い点は解であり, 目的空間に分布している. ここでは解  $\mathbf{x}$  に注目する. 解  $\mathbf{x}$  は, 解  $\mathbf{a}$  と  $\mathbf{b}$  より高い目的関数値を示しており, 式 (2.2) から解  $\mathbf{x}$  は,  $\mathbf{a}$  と  $\mathbf{b}$  を支配するといえる. また, すべての目的関数で解  $\mathbf{x}$  の目的関数値未満の値をとる図中の赤い色付き領域に存在する解は, 解  $\mathbf{x}$  が支配することができる. この領域を解  $\mathbf{x}$  の支配領域という [22]. また, 灰色で示される, すべての目的関数で解  $\mathbf{x}$  の目的関数値以上の値をとる (ただし, 解  $\mathbf{x}$  とすべて同じ目的関数値の場合を除

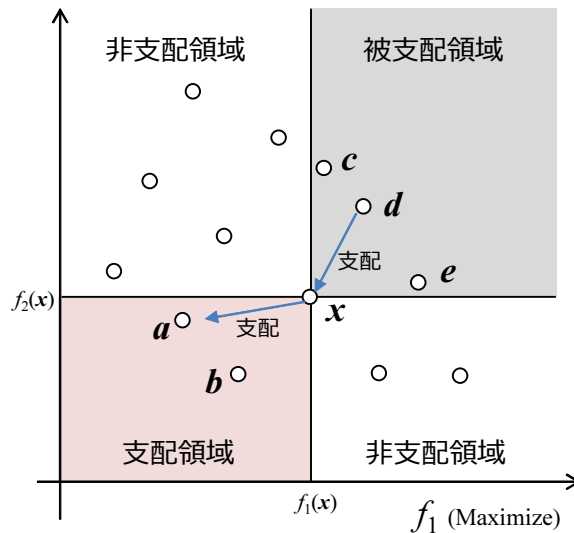


図 2.1: 支配の概念図

く) 領域を解  $x$  の被支配領域という。解  $x$  は、被支配領域に存在する解に支配される。つまり、図中の  $x$  は、 $c$ ,  $d$ ,  $e$  に支配される。目的空間の支配領域と被支配領域以外を非支配領域といい、この領域に存在する解と解  $x$  は非支配の関係になる。

すべての解に支配されない解  $x$  を非劣解という。また、このような非劣解の集合をパレート最適解集合 (Pareto Optimal Solutions: POS) といい、POS が目的空間に作る関数間のトレードオフをパレートフロントという。

## 2.2 進化計算法

進化計算法 (Evolutionary Computation: EC), あるいは進化型アルゴリズム (Evolutionary Algorithm: EA) は、生物進化の原理に基づく計算法で、確率的探索・学習・最適化の方法である [4]。本研究では、EA の一種である遺伝的アルゴリズム (Genetic Algorithm: GA) に着目する。GA は 1975 年に John Holland 氏の “Adaptation in Natural and Artificial Systems” において考案され、頑健な解探索性能が確認されている。自然界において生物は、集団の中で環境に適応しない個体が淘汰され、環境に適応した個体が生き残り子孫を残しやすい。これが繰り返されることにより、より環境に適した個体へと進化していく。GA はこの仕組みをモデル化し、解の持つ変数を生物の遺伝子のように扱い、目的関数を最適化するように解を進化させる。単一目的最適化の場合、単一の最適解を解集団から導出する。そのため、解集団中のある解が最適値に到達すればよい。一方、多目的最適化では、最適パレートフロント面を近似するために、各解をそれぞれパレートフロントの異なる部位へと進化させ、パレート最適解集合を解集団から導出することになる。GA では、

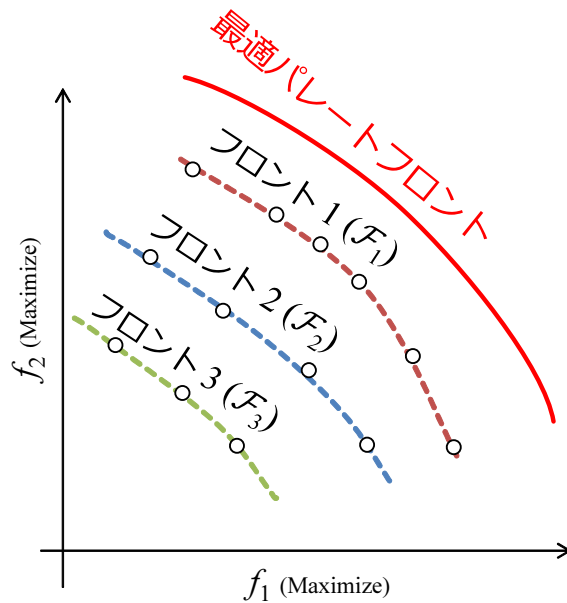


図 2.2: フロント分類の概念図

解の決定変数ベクトル  $\boldsymbol{x} (= x_1, x_2, \dots, x_n)$  を染色体とみなし,  $x_i$  ( $i = 1, 2, \dots, n$ ) は第  $i$  遺伝子座における遺伝子という. 遺伝子表現は 0 と 1 を用いるバイナリ型と実数値を用いる実数値型がある. GA は, ランダムに初期集団となる複数の個体 (解) を生成した後, 以下のプロセスを行う.

- (1) 適応度の算出
- (2) 選択
- (3) 交叉
- (4) 突然変異

これらの操作により, 新しい世代の解集団が作られる. さらに, (1)~(4) のプロセスを繰り返すことによって, 最適な解を探索する.

### 2.2.1 適応度の算出

解集団中のそれぞれの解について適応度を算出する. 適応度は与えられた目的関数に基づいて最適なほど高い値が与えられる. 単一目的最適化問題において, 適応度は単に目的関数値が用いられる. 一方で, 多目的最適化問題においては, 複数の目的関数値を扱うため工夫が必要である. ここでは, 一般的に広く用いられている非支配ソート [14] による解の適応度算出法を説明する.

非支配ソートでは, 解集団  $\mathcal{R}$  中の解を支配されないレベルでフロントと呼ばれる階層



に分類し、ランク付けする。概念図を図 2.2 に示す。具体的には、まず、全解集団  $\mathcal{R}$  から式 (2.2) を用いて非劣解集合を取り出す。この集合を第一フロント  $\mathcal{F}_1$  とする。次に、全解集団から第一フロントを除いた集合  $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{F}_1$  から非劣解集合を取り出し、第二フロント  $\mathcal{F}_2$  とする。以降、全解集団から取り出した非劣解集合を除く解集団から、次のフロントを取り出すことを、すべての解がフロントに分類されるまで続ける。各解の適応度は分類されたフロント  $\mathcal{F}_i$  のインデックス番号  $i$  とし、小さな番号ほど良い解と判断する。この方法によって、より最適なパレートフロントに近い解ほど良い解と判断する。

## 2.2.2 選択

個体の適応度が決定したら、親となる個体を選択する。適応度の高い個体がより多くの子孫を残すように親となる個体を選ぶ。これによって、よりよい個体を形成する遺伝子が集団中に広がる。ただし、適応度の低い親も選択することは、解の多様性を保つ効果がある。代表的な選択方法として GA の分野で広く使用されている比例選択法やトーナメント選択法がある。

- 比例選択法

個体の適応度に比例した確率で選択する。個体  $j (j = 1, 2, \dots, N)$  の適応度を  $F_j$  とすると、個体  $j$  の選択される確率は

$$p_j = \frac{F_j}{\sum_{j=1}^N F_j} \quad (2.3)$$

であり、適応度の高いものほど選択されやすくなる。ここで、この方法は適応度の値がすべて非負かつ大きい値ほど良い場合にのみ適用可能であることに注意する。異なる場合は、選択確率の与え方を適切に変更する必要がある。

- トーナメント選択法

集団からランダムに複数の個体  $t$  個を選び出し、その中から最も高い適応度を持つ個体を選択する。ランダムに選ぶ個体数  $t$  はトーナメントサイズといい、適応度の高い解が選ばれる選択圧を調節することができる。大きな値に設定するほど、解集団内の高い適応度を持つ個体を親として選択しやすくなる。しかし、適応度の高いある個体が何度も親として選択されると、その親の持つ遺伝子情報が次世代の解集団に蔓延することで解の多様性が低下し、局所解に陥る恐れがある。一般に  $t = 2$  に設定されることが多く、これをバイナリトーナメント選択と呼ぶ。

### 2.2.3 交叉

交叉は、ある確率 (交叉率  $P_c$ ) で二つの親の遺伝子を組み替え、子個体を生成する遺伝的操作である。遺伝的操作は遺伝子の表現法により異なる。

#### バイナリ型の交叉法

- 一点交叉

交叉する位置をランダムに一カ所決めてその前と後で、どちらの親の遺伝子を受け継ぐかを変える方法である。図 2.3 では、4 番目と 5 番目の遺伝子座の間に交叉点があり、世代  $t$  における親  $\mathbf{x}^{(1,t)}$  と  $\mathbf{x}^{(2,t)}$  のそれぞれ交叉点前と交叉点後の遺伝子を入れ替え、新しい個体を生成する。

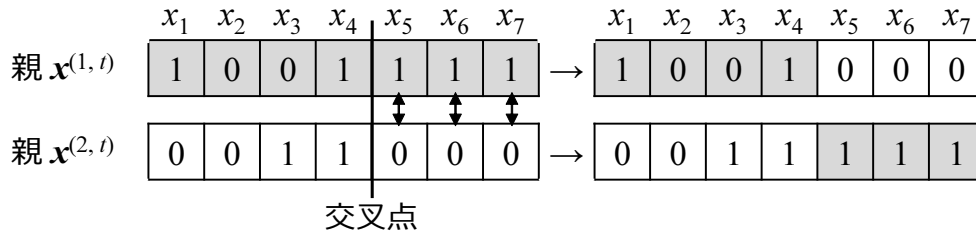


図 2.3: 一点交叉の例

- 複数点交叉

複数点交叉は、交叉位置が複数ある方法である。図 2.4 では二点交叉の例を示しており、2 番目と 3 番目の遺伝子座の間と、5 番目と 6 番目の遺伝子座の間に交叉位置があり、交叉位置で遺伝子を入れ替えをして新しい個体を生成する。

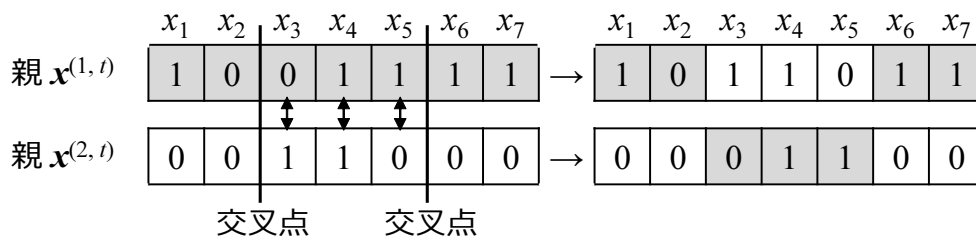


図 2.4: 複数点交叉 (二点交叉) の例

- 一様交叉

一様交叉は、交叉時にマスクをかけて、それによってどちらの親の遺伝子を受

け継ぐかを決める方法である。マスクのビットにはランダムに 0 か 1 を設定する。親とマスクが図 2.5 のような場合、マスクのビットが 0 の時は、子  $\mathbf{x}^{(1,t+1)}$  に親  $\mathbf{x}^{(1,t)}$  の遺伝子をコピーし、マスクのビットが 1 の時は、子  $\mathbf{x}^{(1,t+1)}$  に親  $\mathbf{x}^{(2,t)}$  の遺伝子をコピーする。子  $\mathbf{x}^{(2,t+1)}$  に対してはこの操作の逆を行う。二つの親の各遺伝子座の遺伝子を確率 0.5 で入れ替えることになるため、子はほぼ半分の遺伝子を親  $\mathbf{x}^{(1,t)}$  から、残りを親  $\mathbf{x}^{(2,t)}$  から引き継ぐ。また、マスクは一組の親に対し、ランダムに生成される。

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
親 $\mathbf{x}^{(1,t)}$	1	0	0	1	1	1	1
親 $\mathbf{x}^{(2,t)}$	0	0	1	1	0	0	0
マスク	0	1	0	1	0	0	1
子 $\mathbf{x}^{(1,t+1)}$	1	0	1	1	1	1	0
子 $\mathbf{x}^{(2,t+1)}$	0	0	0	1	0	0	1

図 2.5: 一様交叉の例

### 実数値型の交叉法

Simulated Binary Crossover (SBX) は、各遺伝子  $x_i$  に個別に確率分布を用いた計算を行う方法である。SBX では、子個体の遺伝子は以下の手順で算出する。まず、 $[0,1]$  の乱数  $u_i$  を生成し、それを用いて  $\beta_i$  を算出する。

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & \text{if } u_i \leq 0.5 \\ (\frac{1}{2(1-u_i)})^{\frac{1}{\eta_c+1}} & \text{otherwise} \end{cases} \quad (2.4)$$

次に、世代  $t$  における 2 つの親個体の  $i$  番目の遺伝子  $x_i^{(1,t)}$ ,  $x_i^{(2,t)}$  に対し、 $\beta_i$  を用いて、変異を加える。

$$\begin{aligned} x_i'^{(1,t)} &= 0.5\{(1 + \beta_i)x_i^{(1,t)} + (1 - \beta_i)x_i^{(2,t)}\} \\ x_i'^{(2,t)} &= 0.5\{(1 - \beta_i)x_i^{(1,t)} + (1 + \beta_i)x_i^{(2,t)}\} \end{aligned} \quad (2.5)$$

変異を加えた親個体  $\mathbf{x}'^{(1,t)}$ ,  $\mathbf{x}'^{(2,t)}$  の各遺伝子を、一様交叉と同様に 0.5 の確率で入れ替え、世代  $t+1$  における 2 つの子個体  $\mathbf{x}^{(1,t+1)}$ ,  $\mathbf{x}^{(2,t+1)}$  とする。この方法は親個体同士の各遺伝子が近いほど変異量が小さくなり、親に似た遺伝子を持つ子個体が生成されやすいという特徴がある。また、パラメータ  $\eta_c$  が大きいほど変異量が小さくなりやすく、親の遺伝子に近い値になる確率が高くなる。

## 2.2.4 突然変異

突然変異は，各遺伝子座に対し一定の確率 (突然変異率  $P_m$ ) で遺伝子を変化させる操作である．局所解に陥ることを防ぐ効果や解探索範囲の多様化を促す効果がある．突然変異は，あまり大きな変異確率に設定すると適応度の高い部分解 (スキマタ) が破壊されるためランダムサーチ化してしまうが，ある程度の変異は必要である．一般に  $1/(\text{遺伝子長})$  を突然変異率に採用することが多い．

### バイナリ型の突然変異法

ビット反転突然変異は各遺伝子座に対し，突然変異率に従ってビットを反転させる．図 2.6 にビット反転突然変異の例を示す．第 3 遺伝子座の遺伝子が突然変異を起こして 0 から 1 に変化している．

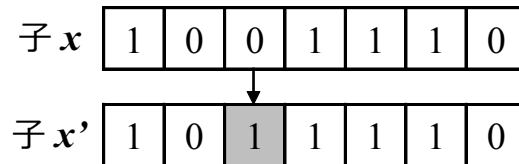


図 2.6: ビット反転突然変異の例

### 実数値型の突然変異法

Polynomial Mutation は，SBX と同様に確率分布を用いた方法である．計算手順は，まず  $[0,1]$  の乱数  $u_i$  を生成し，それを用いて  $\delta_i$  を算出する．

$$\delta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_m+1}} - 1 & \text{if } u_i > 0.5 \\ 1 - \{2(1 - u_i)\}^{\frac{1}{\eta_m+1}} & \text{otherwise.} \end{cases} \quad (2.6)$$

次に，変異させる遺伝子  $x_i$  と変異の上限値  $x_i^{(U)}$ ，下限値  $x_i^{(L)}$ ， $\delta_i$  を用いて変異後の遺伝子  $x'_i$  を算出する．

$$x'_i{}^{(1,t+1)} = x_i^{(1,t+1)} + \delta_i(x_i^{(U)} - x_i^{(L)}) \quad (2.7)$$

パラメータ  $\eta_m$  が大きいほど変異前の遺伝子に近い値になる確率が高くなる．

## 2.3 NSGA-II

ここでは特に多目的最適化問題を解くための多目的進化型アルゴリズム (Multi-Objective Evolutionary Algorithm: MOEA) の代表的なアルゴリズムを説明する．NSGA-II

目的関数値による非支配ソーティング

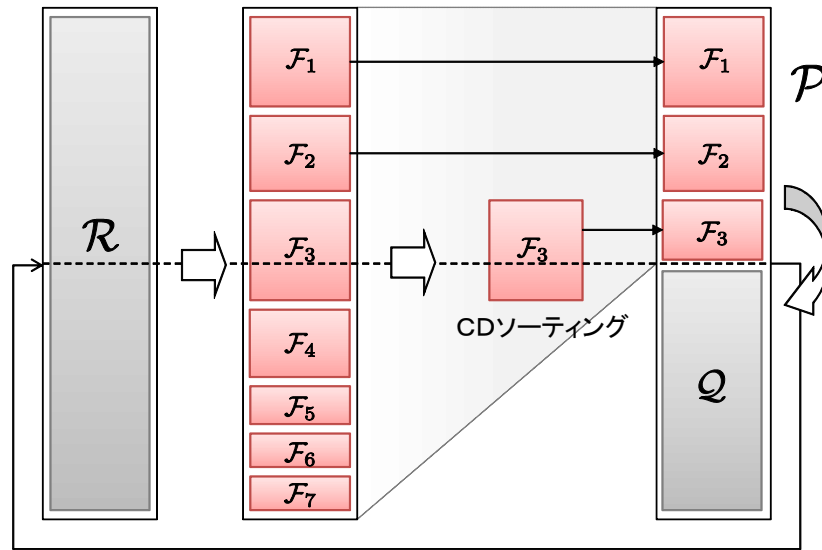


図 2.7: NSGA-II のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団)

(fast Non-dominated Sorting Genetic Algorithm)[14] は式 (2.2) によるパレート支配を利用して親個体を選択するアルゴリズムである。NSGA-II のブロック図を図 2.7 に示す。まず、NSGA-II は、世代  $t$  の解集団  $\mathcal{R}_t$  を、支配されない順にフロントと呼ばれる複数の解集合に分類することによって、解をランキングする。次に、最も支配されない最上位のフロントに属する解から、 $\mathcal{R}_t$  の半数を次世代の親集団  $\mathcal{P}_t$  として選択する。同一のフロント内の解の優劣は、目的空間における分布の混雑度 (Crowding Distance : CD) を考慮して決定する。CD は目的空間における解の混雑度を表し、同一フロントに含まれる個体それぞれについて、各目的関数軸において両隣に位置する個体間の距離を足し合わせた値である。ただし、ある目的関数値について最大、最小の値を持つ個体、つまりフロントの端に位置する解には無限大の値を与える。この値の大きな個体ほど隣り合う解が位置的に遠く混雑度の低い良い解であると判断される。フロントの基準によって、最適なパレートフロントに対する収束性が高い解を解集団から優先的に選択する。また、CD の基準では、まず、それぞれの目的関数の最大値と最小値を有する解を優先的に選択し、次に、目的空間の疎な領域に分布する解を選択する。これにより、解集団が目的空間に広く、均一に分布するようになる。ひとつの子を生成するために、親集団  $\mathcal{P}_t$  からフロントと CD に基づくバイナリトーナメント選択によって親を選択し、交叉と突然変異を施す。これを繰り返して、子集団  $\mathcal{Q}_t$  を生成する。親集団と子集団を結合した  $\mathcal{P}_t \cup \mathcal{Q}_t$  を次世代の集団  $\mathcal{R}_{t+1}$  とし、同様な操作を繰り返して解を進化させる。

## 第 3 章

# 制約付き多目的最適化問題と進化計算による解法

### 3.1 制約付き多目的最適化問題

制約付き多目的最適化問題 (Constrained Multi-objective Optimization Problem: CMOP) は,  $k$  種類の制約条件の下で  $m$  種類の目的関数を最大化 (あるいは最小化) する解  $\mathbf{x}$  を見出す問題である. CMOP は, 以下のように定義される.

$$\begin{cases} \text{Maximize/Minimize} & f_i(\mathbf{x}) & (i = 1, 2, \dots, m) \\ \text{subject to} & g_j(\mathbf{x}) \geq 0 & (j = 1, 2, \dots, k) \end{cases} \quad (3.1)$$

ここで,  $g(\mathbf{x}) \geq 0$  は不等式制約と呼ぶ. 制約条件としてはもう一つ,  $h(\mathbf{x}) = 0$  で表される等式制約が存在する. 等式制約は,  $g(\mathbf{x}) = -|h(\mathbf{x})| \geq 0$  のように不等式制約へ変換することができるため, 本論文では不等式制約のみを扱う.  $k$  種類の全制約条件を満たす解を実行可能解といい, ひとつでも満たせない制約がある解を実行不可能解という. 解  $\mathbf{x}$  の制約違反量ベクトル  $\mathbf{v}(\mathbf{x}) (= \{v_1(\mathbf{x}), v_2(\mathbf{x}), \dots, v_k(\mathbf{x})\})$  は, 次式で定義される.

$$v_j(\mathbf{x}) = \begin{cases} |g_j(\mathbf{x})|, & \text{if } g_j(\mathbf{x}) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, k) \quad (3.2)$$

また, 解  $\mathbf{x}$  の制約総違反量は  $\Omega(\mathbf{x}) = \sum_{j=1}^k v_j(\mathbf{x})$  である.

一方,  $m$  種類の目的関数の間には, 一般的にトレードオフの関係があるため, すべての目的関数を同時に最適化する単一の解は存在しない. そこでパレート支配という概念を用いて解の優劣を決定する. 最大化問題において, 解  $\mathbf{x}$ ,  $\mathbf{y}$  が次式を満たすとき,  $\mathbf{x}$  は  $\mathbf{y}$  を目的関数値において支配する ( $\mathbf{x} \succ_f \mathbf{y}$ ) という.

$$\forall i: f_i(\mathbf{x}) \geq f_i(\mathbf{y}) \wedge \exists i: f_i(\mathbf{x}) > f_i(\mathbf{y}) \quad (i = 1, 2, \dots, m) \quad (3.3)$$

なお, 最小化問題の場合は, 式 (3.3) の不等号を逆にする. また, ある実行可能解  $\mathbf{x}$  について, これを支配する実行可能解が存在しないとき, 解  $\mathbf{x}$  を非劣解という. このような非

劣解の集合をパレート最適解集合 (POS) といい, POS が目的空間に作る関数間のトレードオフをパレートフロントという.

## 3.2 制約付き問題に対する進化計算法

制約なし多目的最適化問題では, すべての解空間において目的を最適化する解を求めるのに対し, 制約付き問題では, 実行可能解の中で目的を最適化する解を求める必要がある. すなわち, 目的関数値のみによって解の適応度を算出することができない問題が生じる. そこでこれまでにいくつかの制約を満たさない実行不可能解の扱い方についての研究がされてきた. 本節では, 進化計算における制約の扱い方について述べる.

### 3.2.1 デスペナルティ法

進化計算で制約条件を有する最適化問題を解く最も簡易な方法として, 生成された実行不可能解を排除するデス・ペナルティ法 [23, 24] がある. これらは単一目的最適化に対して提案されたが, Coello らによって MOEA にも導入されている [25]. この方法では, ランダムに生成した初期集団や, 遺伝的操作によって生成した解が実行不可能になった場合, 解集団への追加を認めない. すなわち, 解集団中の解は, すべて実行可能解になる. この方法では, 制約を満たさない実行不可能解のための特別な処理を進化計算のアルゴリズム中に導入しなくて良い点がメリットであるが, 制約の厳しい問題で実行可能解が得られない場合がある.

### 3.2.2 修復法

生成した実行不可能解を実行可能解へ修復する方法である [26, 27, 28]. デスペナルティ法と同様に, 解集団中の解はすべて実行可能解になるため, 実行不可能解のための特別な処理を進化計算のアルゴリズム中に導入しなくて良いメリットがある. しかし, この方法は, 最適化する問題に対する事前知識があり, 実行不可能解から実行可能解へ変更する方法が既知の場合にのみ適用できる. そのため, この方法は, すべての制約付き最適化問題に適用できるわけではない. また, 制約付き最適化問題ごとに修復処理を設計する必要がある.

### 3.2.3 ペナルティ法

実行不可能解を解集団に含めることを認め, 制約違反量とペナルティ係数に応じて目的関数値を変更した適応度に基づいて解集団を進化させるアプローチとしてペナルティ法が挙げられる. この方法では, 制約違反量の大きな解ほど目的関数値へのペナルティ

が大きくなり、適応度を低くする。これまで単一目的最適化のために検討されてきた全解探索を通じて固定のペナルティ係数を用いる静的ペナルティ法 [29]、解探索の進行に伴ってペナルティ係数を変化させる動的ペナルティ法 [30]、解集団の状態に応じてペナルティ係数を変化させる適応的ペナルティ法 [31] のそれぞれが、CMOP へ展開されている [32, 33, 34, 35]。CMOP の場合、解が有する目的関数値が複数になるため、それぞれの目的関数値を制約総違反量に応じて変化させた適応度を算出する。ペナルティ法では、ペナルティ係数を考慮した制約違反量を、目的関数値に合成することによって適応度を算出するため、適切なペナルティ係数は、最適化する問題に依存することが知られている [36]。

### 3.2.4 目的関数値と制約違反量の個別評価法

目的関数値と制約違反量を合成するペナルティ法に対して、目的関数値と制約違反量をそれぞれ別々に取り扱い、実行不可能解を可能解へ進化させる方法がある [36]。CMOP の解法においては、複数の目的関数値と、複数の制約違反量の足し合わせである制約総違反量  $\Omega$  に基づいて解集団を進化させる方法が提案されている [14, 37]。また、複数の制約違反量にパレート支配の概念を導入し、実行不可能解を制約違反量に関して支配されないレベルで優劣決定する方法が提案されている [21, 38, 39]。

このアプローチの代表的手法である CNSGA-II (Constrained NSGA-II) [14] と Ray らの RTS アルゴリズム [21]、C-MOEA/D (Constraint Multi-Objective Evolutionary Algorithm on Decomposition) [41] について詳細に述べる。

#### CNSGA-II

CNSGA-II は、NSGA-II を CMOP の解法に拡張したアルゴリズムで、式 (2.2) の目的関数値による支配の代わりに、総制約支配 [14] を用いる。以下のいずれかを満たすとき、解  $\mathbf{x}$  は  $\mathbf{y}$  を総制約支配する ( $\mathbf{x} \succ_{\Omega} \mathbf{y}$ ) という。

1.  $\mathbf{x}$  が実行可能解で、 $\mathbf{y}$  は実行不可能解である
2.  $\mathbf{x}$  と  $\mathbf{y}$  が実行不可能解で、 $\mathbf{x}$  の方が制約総違反量が少ない ( $\Omega(\mathbf{x}) < \Omega(\mathbf{y})$ )
3.  $\mathbf{x}$  と  $\mathbf{y}$  が実行可能解で、 $\mathbf{x}$  が  $\mathbf{y}$  を目的関数値で支配する ( $\mathbf{x} \succ_f \mathbf{y}$ )

CNSGA-II のブロック図を図 3.1 に示す。CNSGA-II は、全解集団  $\mathcal{R}$  を総制約支配されない順にフロントと呼ばれるいくつかの階層 ( $\mathcal{F}_1, \mathcal{F}_2 \dots$ ) に分類する。上位フロントから順に、解集団  $\mathcal{R}$  の半数を次世代の親集団  $\mathcal{P}$  とする。この際、同一フロントに属する解の優劣は、目的空間における解の混雑度距離 (Crowding Distance, 以下 CD) を考慮して決定する。これにより、解集団中の実行可能解は、NSGA-II と同様に目的関数値において支配 ( $\succ_f$ ) されない順に複数のフロントへ分類される。実行不可能解は、可能解より低いフロントになり、制約総違反量が小さい順にランキングされる。



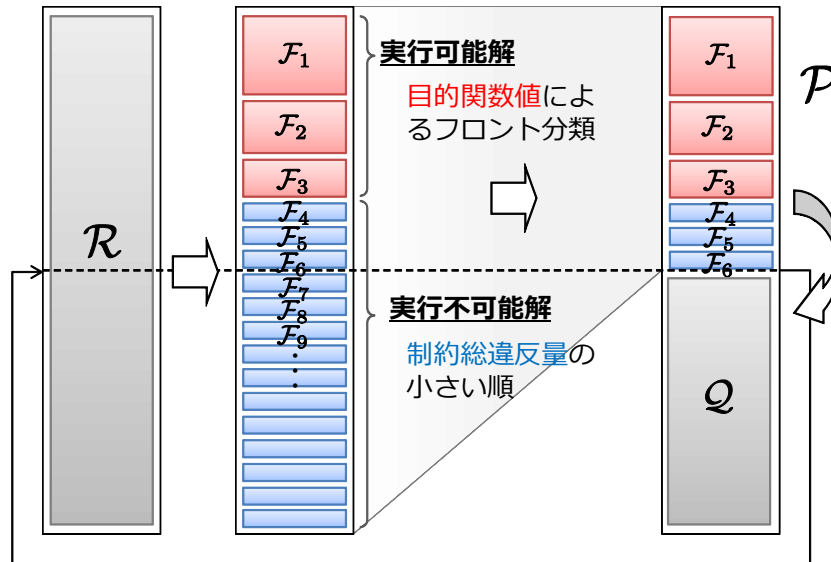


図 3.1: CNSGA-II のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団)

CNSGA-II は、各制約違反量の和  $\Omega$  で実行不可能解を評価するため、各制約違反量のスケールの違いが解の進化に影響することがある。また、実行可能解を獲得する過程では、制約総違反量  $\Omega$  のみが考慮されるため、結果として目的関数値の不良な実行可能解が獲得されることがある。さらに、実行可能解が不可能解より優先的に子を生成するため、実行不可能解が持つ有用な遺伝子が解探索に生かされにくい点に課題がある。

## RTS

Ray らは、三種類の非支配ランクを用いる MOEA (以下、RTS と称す) を提案している [21]. 式 (2.2) による  $m$  種類の目的関数値に基づいた非支配ソートによって分類されたフロント  $\mathcal{F}^{\text{obj}}$  によるランク,  $k$  種類の制約違反量によって分類されたフロント  $\mathcal{F}^{\text{con}}$  によるランク, そして、目的関数値と制約違反量を合成した  $m+k$  種類の要素によって分類されたフロント  $\mathcal{F}^{\text{com}}$  によるランクである。RTS のブロック図を図 3.2 と図 3.3 に示す。各世代において、RTS は目的関数値と制約違反量に基づくフロントの最上位 ( $=\mathcal{F}_1^{\text{com}}$ ) に属す解のうち、実行可能解を次世代の解集団にコピーする。これは実行可能解の中で非劣解を選出するのと同義である。子を生成するために、解集団から親  $A$  と親候補  $B, C$  を選択する。親  $A$  の選択法には、制約の易しい問題のために RTS-O と、制約の厳しい問題のために RTS-C の二種類がある。RTS-O では、 $\mathcal{F}^{\text{obj}}$  のランクに基づく比例選択によって親  $A$  を選ぶ。一方、RTS-C では、 $\mathcal{F}^{\text{con}}$  のランクに基づく比例選択によって親  $A$  を選ぶ。親候補  $B$  と  $C$  は、RTS-O と RTS-C 共に、 $\mathcal{F}^{\text{con}}$  のランクに基づく比例選択によって選ぶ。以下の基準で親候補  $B$  と  $C$  を比較し、親  $A$  と交配する相手を決定する。

$B$  と  $C$  が実行可能のとき:

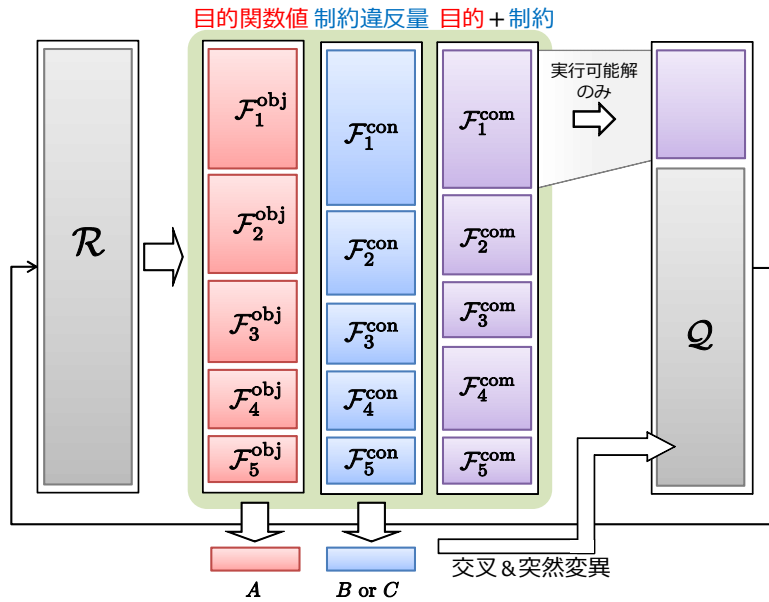


図 3.2: RTS-O のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{Q}$ : 子集団)

$B$  と  $C$  のうち,  $\mathcal{F}^{\text{obj}}$  のランク番号が小さい方を親にする. 同フロントに属す場合は, ニッチカウント [21] により, 目的空間における混雑度が低い方を親にする.

**$B$  と  $C$  が実行不可能のとき:**

$B$  と  $C$  のうち,  $\mathcal{F}^{\text{con}}$  のランク番号が小さい方を親にする. 同フロントに属す場合は, 親  $A$  と共通して満足している制約の数が少ない方を親にする [21].

**$B$  と  $C$  の一方が実行可能, 一方が実行不可能のとき:**

$B$  と  $C$  のうち, 実行可能の方を親にする.

RTS は, 制約違反量に基づく非支配ランクを導入することによって, 各制約違反量のスケールの違いを考慮しなくて良いところに利点がある. しかし, RTS-O では, 親  $A$  が  $\mathcal{F}^{\text{obj}}$  のランクのみに基づいて選択されるため, 制約違反量が高い不良な解が親に選ばれることがある. 逆に, RTS-C では, 親  $A$  が  $\mathcal{F}^{\text{con}}$  のランクのみに基づいて選択されるため, 目的関数値が不良な解が親に選ばれることがある.

他にも, 制約違反量による支配を導入する方法として, 差分進化に基づいて関数最適化する GDE (Generalized Differential Evolution) [38] や, 解集団中の実行不可能解集合に対して制約違反量に基づく支配を適用する Oyama らの方法 [39] がある. これらの方法は, 多目的問題を考慮したアルゴリズムに設計されているが, 文献 [38, 39] における性能検証が単一目的問題によるものであるため, 本研究では, これらを取り上げず, RTS を制約違反量による非支配ランクを用いる MOEA として, 提案法の比較アルゴリズムとする.

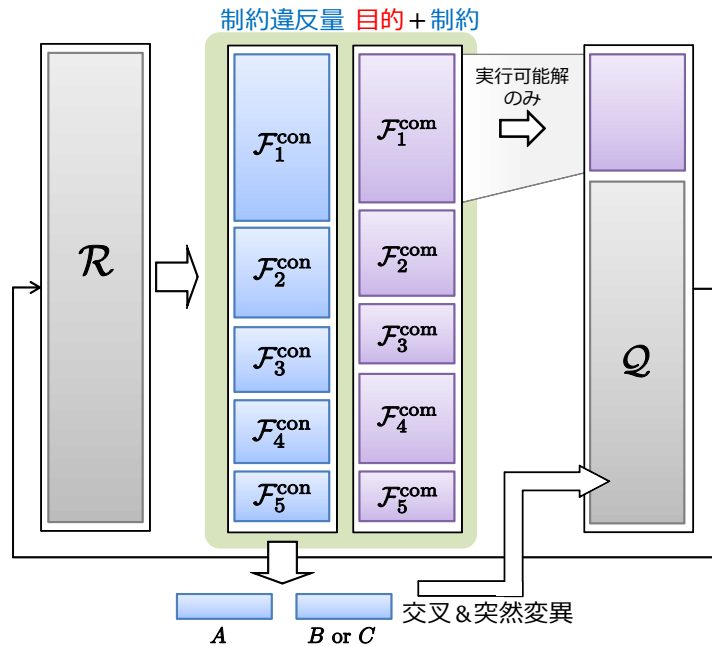


図 3.3: RTS-C のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{Q}$ : 子集団)

### C-MOEA/D

NSGA-II に続いて代表的な MOEA である MOEA/D (Multi-Objective Evolutionary Algorithm on Decomposition)[40] を制約付き問題の解法に発展させた Constraint-MOEA/D (C-MOEA/D) [41] がある. MOEA/D は, 多目的最適化問題を多数の単一目的最適化問題に分解して解く. MOEA/D の概念図を図 3.4 に示す. 一つひとつの単一目的最適化問題は, 目的空間に均一分布する重みベクトル群  $\lambda^1, \lambda^2, \dots, \lambda^N$  に基づくスカラー化関数  $g(\mathbf{f}|\lambda^i)$  ( $i = 1, 2, \dots, N$ ) によって定義される. 各重みベクトル  $\lambda^i$  にひとつの解  $\mathbf{x}^i$  が割り当てられ, サイズ  $N$  の解集団を構成する. 子を生成するとき, MOEA/D は, ひとつの重みベクトル  $\lambda^i$  に注目し, その近傍  $T$  の重みベクトル群に割り当てられた解集合から二つの親を選択し, 交叉と突然変異を施して子  $\mathbf{y}$  を生成する. 図 3.4 の例では,  $\lambda^5$  に注目し, その  $T = 3$  近傍の解から親を選び, 子  $\mathbf{y}$  が生成される例を示している. このように, 目的空間で近い解を交叉することによって, 解集団中に存在する変数情報が乖離した解を交叉する非効率な遺伝的操作を回避することができる. 次に, 生成された子  $\mathbf{y}$  によって, 解集団中の解  $\mathbf{x}^i$  の更新を試みる.

MOEA/D では単純に生成された子  $\mathbf{y}$  が解  $\mathbf{x}^i$  のスカラー化関数値が良いとき, 解を子で更新する. C-MOEA/D では, 以下の各条件で  $\mathbf{x}^i$  の更新を試みる.

1. 解  $\mathbf{x}^i$  が実行可能で, 子  $\mathbf{y}$  が実行不可能のとき,  $\mathbf{y}$  は  $\mathbf{x}^i$  を更新しない.
2. 解  $\mathbf{x}^i$  が実行不可能で, 子  $\mathbf{y}$  が実行可能のとき,  $\mathbf{y}$  は  $\mathbf{x}^i$  を更新する.

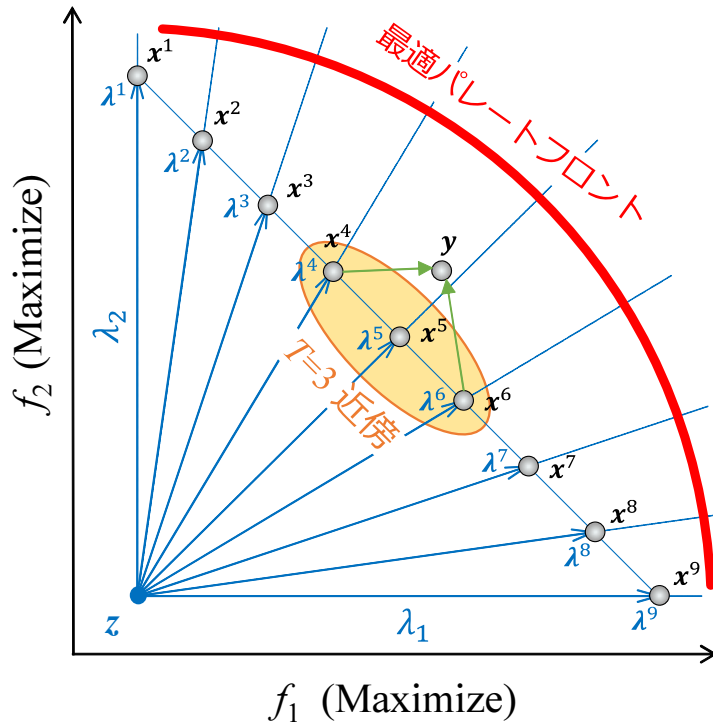


図 3.4: MOEA/D の概念図

3. 解  $x^i$  と子  $y$  が実行不可能で、 $\Omega(x^i) > \Omega(y)$  のとき、 $y$  は  $x^i$  を更新する。
4. 解  $x^i$  と子  $y$  が実行可能で、 $g(\mathbf{f}(y)|\lambda^i)$  が  $g(\mathbf{f}(x^i)|\lambda^i)$  がより良いとき、 $y$  は  $x^i$  を更新する。

C-MOEA/D は、実行不可能解を制約総違反量  $\Omega$  で比較するため、実行可能解を獲得する過程で目的関数値が考慮されない。この特徴は、CNSGA-II と一致する。また、目的関数値ベクトル  $\mathbf{f}$  をスカラー化するため、各目的関数値のスケーリングによって、解探索性能が左右する特徴がある。

### 3.3 本研究の位置づけ

上記の従来法を、まず、解集団が実行可能解のみを含むか、実行可能解と実行不可能解の両方を含むかで表 3.1 に分類した。解集団が実行可能解のみを含む表 3.1 の **A** に分類される方法は、進化計算のアルゴリズムに実行不可能解を扱うための特別な処理を必要としない利点がある。しかし、修復法は、対象とする最適化問題に対する事前知識があるごく限られた場合にしか適用できない。また、デスペナルティ法は、生成した解が実行不可能なら捨ててしまい、実行可能解しか解集団に認めないため、実行可能解が得られにくい問題

表 3.1: 制約付き最適化問題の進化計算による解法の第一分類

		アプローチ
解 集 団	実行可能解 のみ	<b>A:</b> デスペナルティ法 修復法
	実行可能解と 実行不可能解	<b>B:</b> ペナルティ法, CNSGA-II, RTS-O, RTS-C, C-MOEA/D, 本研究

表 3.2: 分類 **B** の第 2 フェーズにおける細分類

		実行不可能解を親として選択 しない		する
親の選択領域	なし	<b>B-1:</b> ペナルティ法 CNSGA-II RTS-C	<b>B-3:</b> RTS-O	
	あり	<b>B-2:</b> C-MOEA/D	<b>B-4:</b> 本研究	

では十分に最適化できず、現実的なアプローチではない。一方、解集団に実行可能解と実行不可能解の両方を含む表 3.1 の **B** に分類される方法は、進化計算のアルゴリズムに実行不可能解を扱うための特別な処理を導入し、実行不可能解を実行可能解へ進化させる。実行可能解が得られにくい問題においても、制約違反量を最小化するように実行不可能解を進化させることによって実行可能解の獲得を試みるため、デスペナルティ法より有望なアプローチである。本研究では、表 3.1 の **B** に該当する方法を構築する。

表 3.1 の **B** に分類される方法による解探索過程は、大きく 2 つのフェーズにわかれる。第 1 フェーズでは、解集団中の実行不可能解の制約違反量を最小化して実行可能解を獲得することを目的とする。第 2 フェーズでは、実行可能解の目的関数値を高めることを目的とする。**B** に分類される方法はいずれも、実行不可能解を扱うための特別な処理を導入

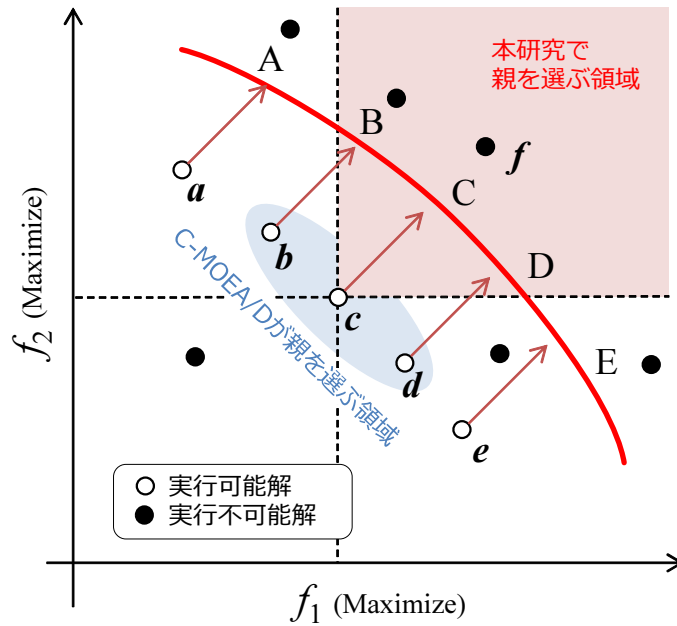


図 3.5: 親の選択領域と解の探索方向

するが、第1フェーズにおける実行不可能解の扱いは、ほぼ同様で、制約違反量が小さな実行不可能解ほど子を生成する機会を与える。実行不可能解の扱い方に違いが生じるのは、第2フェーズである。

第2フェーズにおける表 3.1 の B の方法を、解集団中に実行可能解が存在するのにも関わらず実行不可能解を親に選択するかしないか、親選択において考慮する目的空間の領域があるかないかで表 3.2 に分類した。

まず、実行不可能解を親として選択しないか、するかの違いについて述べる。実行不可能解を親として選択しない方法を、表 3.2 の B-1, B-2 に示す。これらの方法は、解集団中に獲得した実行可能解を親として子生成し、実行不可能解は活用せずに解集団から削除する。進化計算は、広い変数空間を限られた数の解集団で解探索する。実行可能解のみを用いるとなると、限られた解集団の中のさらに一部の情報源のみを用いて探索することになる。一方、実行不可能解を親として選択する方法を B-3, B-4 に示す。これらの方法は、解集団に含まれる全ての解の情報を利用することになる。実行不可能解の中には、制約条件は満たさないものの実行可能解より良い目的関数値を持つ解があるため、それらを解探索に活用できる可能性がある。

次に、親選択において考慮する目的空間の領域がないか、あるかの違いについて述べる。多目的最適化の過程では、パレートフロントを近似するために多様な実行可能解を解集団に維持するようになる。2目的問題における解集団と最適パレートフロントを図 3.5 に例示した。目的関数値がかけ離れた解 a と e は、一般的に、遺伝子(変数)も大きく乖

離している。これらの交配は、親の有する遺伝子構造を著しく破壊することにつながり、解探索が非効率化する恐れがある。親選択において考慮する目的空間の領域がない方法を **B-1**, **B-3** に示す。これらの方法は、目的空間における解の位置を考慮せずに親を選択する。この場合、目的関数値がかけ離れた解  $a$  と  $e$  が交配される可能性があり、解探索が非効率化する恐れがある。親選択において考慮する目的空間の領域がある方法は、実行可能解を親とする **B-2** と実行不可能解も親にする **B-4** に分かれる。まず、実行可能解を親とする **B-2** に分類される C-MOEA/D は、目的関数値に近い実行可能解を親のペアとする。図 3.5 に、実行可能解の親  $c$  と交配できる解の範囲を示した。このように、目的関数値に近い解を交配することによって、変数が大きく乖離した解の交配を避け、解探索の効率化を狙う。これに対して、実行不可能解を親の対象にできると、実行可能解より目的関数値が良い実行不可能解を解探索の手がかりとして活用できる可能性がある。これが、本論文で検討する方法で、**B-4** に該当する。多目的最適化では、解集団中のそれぞれの解をパレート最適解へ進化させたい。図 3.5 中の解  $a$  は  $f_2$  が高く、解  $e$  は  $f_1$  が高い。そのため、解  $a$  は、パレートフロントの領域 E より領域 A の近似を目指すほうが良い。逆に、解  $e$  は、領域 A より領域 E の近似を目指すほうが良い。このように、それぞれの解は、近似すべきパレートフロントの領域は異なるものの、いずれの解も、その全ての目的関数値を同時に高める方向へ進化することが望ましいところに共通点がある。本論文では、ある解について、その全ての目的関数値を同時に高める方向を、探索すべき方向とする。図 3.5 中の実行可能解の探索すべき方向を赤の矢印で示した。 $c$  の探索すべき方向には、 $f$  が存在することがわかる。 $f$  は実行不可能解だが、 $c$  より目的関数値が高いため、 $c$  を探索すべき方向に進化させる変数情報を有する可能性がある。本研究では、これらを交配することによって、 $c$  の探索すべき方向への進化を促す。C-MOEA/D のように、 $c$  を  $b$  や  $d$  と交配するより、 $f$  の変数情報を活用したほうが、 $c$  を探索すべき方向へ進化できる可能性がある。このように、実行不可能解を親とし、さらに、それぞれの解の探索すべき方向を尊重した目的空間における親の選択領域を考慮する表 3.2 の **B-4** に該当する方法は、これまでに例がない。これらの双方を考慮することにより、解探索性能を高めることを狙い、**B-4** に該当する上記のコンセプトを実現する方法を構築する。

## 第 4 章

# テスト問題と評価尺度

### 4.1 テスト問題の扱い

進化計算は、最適化問題をブラックボックスとして扱い、それぞれの問題に対する事前知識を利用せずに最適化可能なところに利点がある。そのため、本論文では、複数のテスト問題を用いるが、それぞれの問題の特徴や事前知識は利用せずに最適化する。

### 4.2 離散テスト問題

離散値の変数空間を有するテスト問題として、本研究では、 $m$  目的  $k$  ナップザック問題 ( $mk$ -KP) [15] を用いる。

#### 4.2.1 $m$ 目的 $k$ ナップザック問題 ( $mk$ -KP)

$mk$ -KP は、目的数  $m$  と制約数  $k$  を独立して設定できる点が、Zitzler らが提案する多目的 0/1 ナップザック問題 [26] と異なる。 $mk$ -KP は、次式で定義される。

$$\begin{cases} \text{Maximize} & f_i(\mathbf{x}) = \sum_{l=1}^n p_{li} \cdot x_l \quad (i = 1, 2, \dots, m) \\ \text{subject to} & \sum_{l=1}^n w_{lj} \cdot x_l \leq c_j \quad (j = 1, 2, \dots, k) \end{cases} \quad (4.1)$$

$mk$ -KP の概念図を図 4.1 に示す。 $mk$ -KP には、 $n$  個のアイテムと  $k$  個のナップザックがあり、それぞれのアイテム  $l$  は、 $m$  個の価値  $p_{li}$  ( $i = 1, 2, \dots, m$ ) と、 $k$  個の重さ  $w_{lj}$  ( $j = 1, 2, \dots, k$ ) を持つ。この問題は、 $k$  種類のナップザックの容量を超過しない範囲で、 $m$  種類の価値の総和を最大化するアイテムの組み合わせを表す解  $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in \{0, 1\}^n$  を見出す問題である。本研究では、アイテム数は  $n = 500$  とし、文献 [26] に倣って価値と重さは  $[10, 100]$  の範囲のランダムな整数値に設定する。各ナップザックの容量  $c_j$  は、次式で定義される。

$$c_j = \phi_j \cdot \sum_{l=1}^n w_{lj} \quad (j = 1, 2, \dots, k) \quad (4.2)$$



$$\mathbf{x} = (x_1, \dots, x_l, \dots, x_n) \quad x_i \in \{0,1\}$$

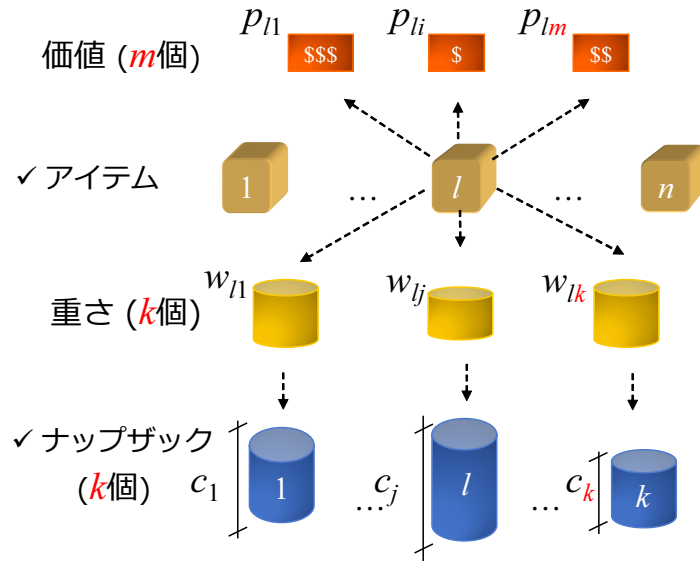


図 4.1:  $m$  目的  $k$  ナップザック問題 ( $mk$ -KP)

$\phi_j$  は、各ナップザックの実行可能率で、取りうる範囲は  $[0.0, 1.0]$  である。  $\phi_j$  が大きいほど、変数空間中に占める実行可能解が多くなるため、制約条件を満足することが易しい問題になる。逆に  $\phi_j$  が小さいほど、変数空間中に占める実行可能解が少なくなるため、制約条件を満足することが厳しい問題になる。本研究では、すべてのナップザックに対して共通の  $\phi$  ( $= \phi_1 = \phi_2 = \dots = \phi_k$ ) を用いる。また、 $m$  目的  $k$  ナップザック問題における制約違反量は次式で定義される。

$$v_j(\mathbf{x}) = \begin{cases} \sum_{l=1}^n w_{lj} \cdot x_l - c_j & , \quad \sum_{l=1}^n w_{lj} \cdot x_l > c_j \\ 0 & , \quad \text{otherwise.} \end{cases} \quad (j = 1, 2, \dots, k) \quad (4.3)$$

$mk$ -KP は、変数空間サイズ  $n$  や目的数  $m$ 、制約数  $k$ 、実行可能率  $\phi$  を設定可能であるため、これらの問題のパラメータを変化させた場合の解探索性能について議論できるところに利点がある。

ここで、 $mk$ -KP において、制約数  $k$  と実行可能率  $\phi$  に対する実行可能解の数の変化について調査する。実験のパラメータは目的数  $m = 2$ 、アイテム数  $n = 20$ 、制約数  $k = \{1, 3, 5, \dots, 21\}$ 、実行可能率  $\phi = \{0.25, 0.4, 0.5, 0.65, 0.75\}$  を用い、解空間すべての  $2^{20}$  種類の解の組み合わせを全探索して実行可能解を求めた。なお結果はアイテムの重さ  $w_{il}$  に依存するため、各試行においてアイテムの重さをそれぞれ  $[10, 100]$  の範囲の整数値でランダムに決定し、30 回の独立した試行の平均値を用いた。得られた実行可能解数の結果を図 4.2 に示す。この結果から、制約数  $k$  の増加に伴って、解空間中に占める実行可能解の数が減少することがわかる。しかし、その影響は制約数  $k$  の増加に線形でなく、増

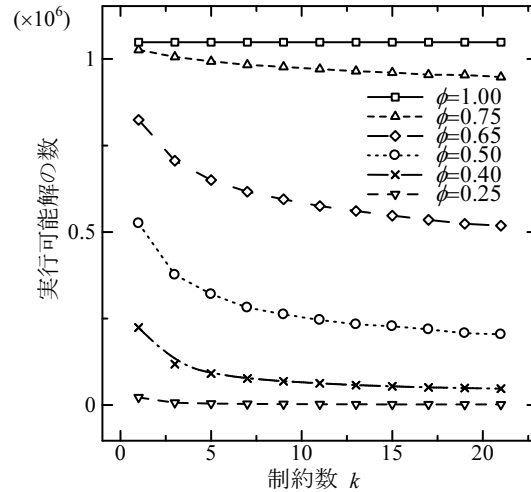


図 4.2:  $mk$ -KP における制約数  $k$ , 実行可能率  $\phi$  に対する実行可能解数の変化

加に伴って緩やかになる。各制約条件の実行不可能領域は独立でなく共通部分があるためである。また、また実行可能率  $\phi$  が小さくなるほど実行可能解の数が減少し、実行可能領域が制限される難しい問題となることがわかる。また、解の実行可能領域に対して実行可能率が制約数よりも大きな影響を与えることがわかる。このように制約数  $k$  が多く、実行可能率  $\phi$  が小さい問題においては、MOEA による解探索の過程で遺伝的操作によって生成される解が制約条件を満たさない実行不可能解になる可能性が高くなる。そのため、実行不可能解が有する遺伝子情報を手がかりにし、多発する実行不可能解を如何に解探索に活かすかが重要な鍵になると考えられる。

## 4.3 連続テスト問題

本研究では、連続問題として SRN [16], TNK [17], OSY [18], mCDTLZ [19] を用いる。それぞれの特徴として、各問題における目的数  $m$ , 制約数  $k$ , 実行可能率, パレートフロントの形状, 変数空間の実行可能領域におけるパレート最適解集合 (POS) の存在位置について表 4.1 に示す。ここで, SRN, TNK, OSY 問題における実行可能率は文献 [35] に掲載された値である。mCDTLZ 問題は, 目的数  $m$  (=制約数  $k$ ) にスケラビリティがあり, 自由に設定できる。また, POS の位置については, POS が実行可能領域の内部に存在する場合と, 実行可能領域と実行不可能領域の境界に存在する場合にわかれる。

### 4.3.1 SRN

SRN は, 凸型のパレートフロントを有する関数最適化問題である。また, 変数空間の実行可能領域の内部にパレート最適解集合が存在する特徴を有する。SRN は以下の式に

表 4.1: テスト問題のパラメータ

	目的数 $m$	制約数 $k$	実行可能率	パレート フロント形状	POS の位置
SRN [16]	2	2	0.1618	凸	内部
TNK [17]	2	2	0.0509	非連続	境界
OSY [18]	2	6	0.0325	凸	境界
mCDTLZ [19]	任意 (ただし $m = k$ )	任意 (ただし $m = k$ )	0.6591 ( $m = k = 2$ の場合)	凹	境界

よって定義される.

$$\left\{ \begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\ \quad \quad \quad f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2 \\ \text{Subject to } g_1(\mathbf{x}) = x_1^2 + x_2^2 \leq 225 \\ \quad \quad \quad g_2(\mathbf{x}) = x_1 - 3x_2 + 10 \leq 0. \end{array} \right. \quad (4.4)$$

解 (変数ベクトル)  $\mathbf{x}$  は 2 変数  $\{x_1, x_2\}$  からなり, それぞれ  $-20 \leq x_1, x_2 \leq 20$  の範囲の実数値をとる.

### 4.3.2 OSY

OSY は, 凸型のパレートフロントを有する関数最適化問題である. また, 変数空間の実行可能領域と実行不可能領域の境界にパレート最適解集合が存在する特徴を有する. OSY は以下の式によって定義される.

$$\left\{ \begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = -[25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2] \\ \quad \quad \quad f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \\ \text{Subject to } g_1(\mathbf{x}) = x_1 + x_2 - 2 \geq 0 \\ \quad \quad \quad g_2(\mathbf{x}) = 6 - x_1 - x_2 \geq 0 \\ \quad \quad \quad g_3(\mathbf{x}) = 2 - x_2 + x_1 \geq 0 \\ \quad \quad \quad g_4(\mathbf{x}) = 2 - x_1 - 3x_2 \geq 0 \\ \quad \quad \quad g_5(\mathbf{x}) = 4 - (x_3 - 3)^2 - x_4 \geq 0 \\ \quad \quad \quad g_6(\mathbf{x}) = (x_5 - 3)^2 + x_6 - 4 \geq 0 \end{array} \right. \quad (4.5)$$

解 (変数ベクトル)  $\mathbf{x}$  は 6 変数  $\{x_1, x_2, \dots, x_6\}$  からなり, それぞれ  $0 \leq x_1, x_2, x_6 \leq 10$ ,  $1 \leq x_3, x_5 \leq 5$ ,  $0 \leq x_4 \leq 6$  の範囲の実数値をとる.

### 4.3.3 TNK

TNK は, 非連続のパレートフロントを有する関数最適化問題である. また, 変数空間の実行可能領域と実行不可能領域の境界にパレート最適解集合が存在する特徴を有する.

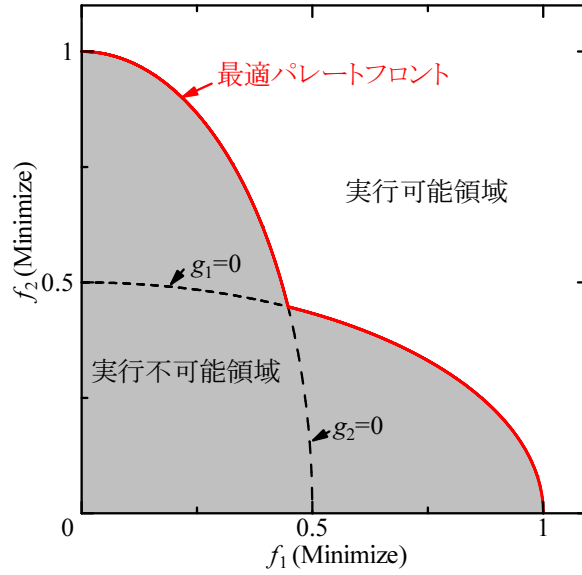


図 4.3: mCDTLZ 問題の目的空間 ( $m = 2$  目的)

TNK は以下の式によって定義される.

$$\begin{cases} \text{Minimize } f_1(\mathbf{x}) = x_1 \\ \quad \quad \quad f_2(\mathbf{x}) = x_2 \\ \text{Subject to } g_1(\mathbf{x}) = x_1^2 + x_2^2 - 1 - 0.1 \cos(16 \arctan \frac{x_1}{x_2}) \geq 0 \\ \quad \quad \quad g_2(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5. \end{cases} \quad (4.6)$$

解 (変数ベクトル)  $\mathbf{x}$  は 2 変数  $\{x_1, x_2\}$  からなり, それぞれ  $[0, \pi]$  の範囲の実数値をとる.

#### 4.3.4 mCDTLZ

mCDTLZ [19] は  $m$  目的  $m$  制約の関数最適化問題である. パラメータ  $m$  によって目的数と制約数を変更可能である特徴がある. mCDTLZ は以下の式によって定義される.

$$\begin{cases} \text{Minimize } f_i(\mathbf{x}) = \frac{1}{\lfloor \frac{n}{m} \rfloor} \sum_{l=\lfloor (i-1)\frac{n}{m} \rfloor}^{\lfloor i\frac{n}{m} \rfloor} x_l^{0.5} & (i = 1, 2, \dots, m) \\ \text{Subject to } g_i(\mathbf{x}) = f_i(\mathbf{x})^2 + 4 \sum_{l=1, l \neq i}^m f_l(\mathbf{x})^2 - 1 \geq 0 & (i = 1, 2, \dots, m) \end{cases} \quad (4.7)$$

解 (変数ベクトル)  $\mathbf{x}$  は  $n$  個の変数  $\{x_1, x_2, \dots, x_n\}$  からなり, それぞれ  $[0, 1]$  の範囲の実数値をとる. 本研究では, 変数の数  $n = 12$  として使用する. 各制約関数が  $g_i = 0$  ( $i = 1, 2, \dots, m$ ) となる境界面の一部がパレートフロントとなる. 目的の数を定めると, 同時に制約数も同じになる点に留意する. 2 目的の mCDTLZ における目的空間を図 4.3 に示す. 図中の影付きの領域は実行不可能領域を示しており, パレートフロントは実行可能領域と実行不可能領域の境界に位置していることがわかる.

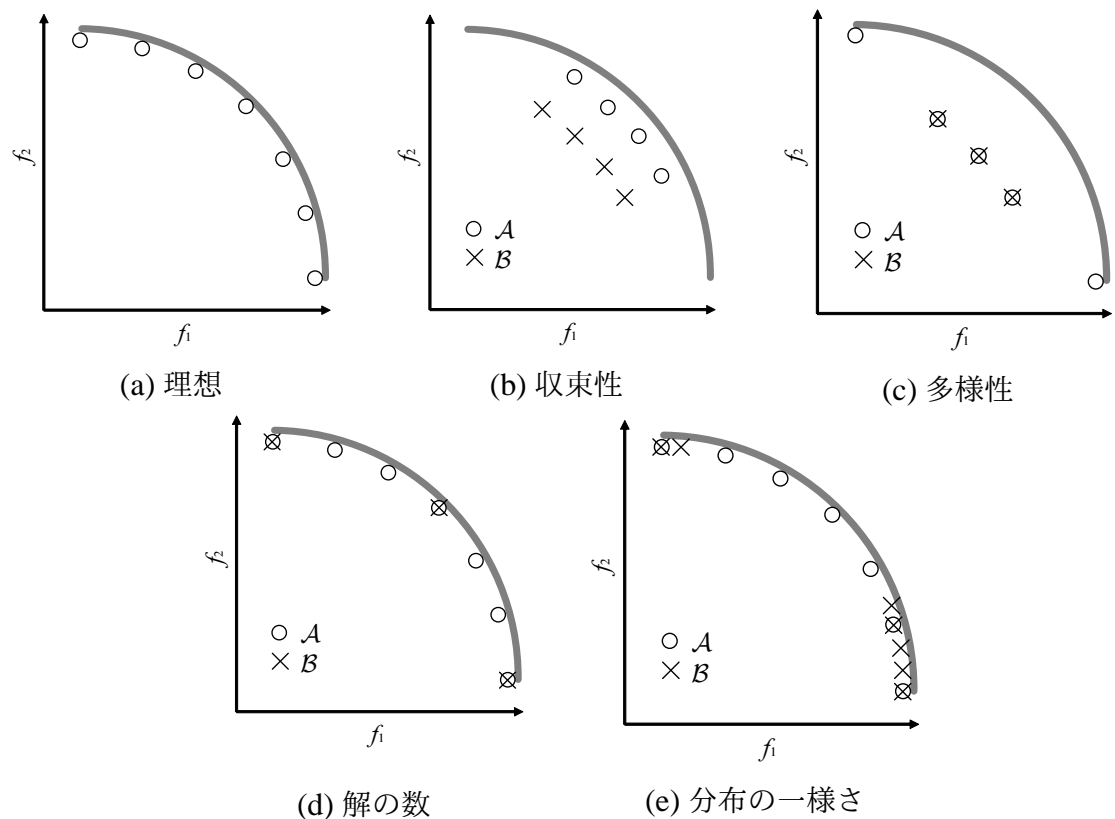


図 4.4: 獲得した解集合とその評価

### 4.3.5 3 段式ハイブリッドロケットの概念設計問題

連続問題のひとつとして、実問題の 3 段式ハイブリッドロケットの概念設計問題 [20] を用いる。この問題は、ロケットの設計が 25 種類の実数の設計変数によって決定される。この実数の設計変数ベクトルを、軌道投入能力及び精度、ノズル開口比、ロケットの縦横比等からなる全 15 の制約条件のもとで、ロケット全備質量 ( $f_1$ ) の最小化とペイロード質量比 ( $f_2$ ) の最大化の 2 目的最適化する。この問題の評価モジュールは、航空宇宙機の設計を専門とする首都大学東京の金崎雅博准教授より提供されたものである。本研究では、この評価モジュール内の処理を考慮せず、ブラックボックスとして扱う。他のテスト問題と比較して、この問題は、解の評価シミュレーションに時間を要するところに特徴がある。

## 4.4 評価尺度

進化計算による多目的最適化問題の解探索におけるゴールは、図 4.4(a) に示すように、与えられた問題の真のパレートフロントを限られた解集合によって高精度に近似するこ

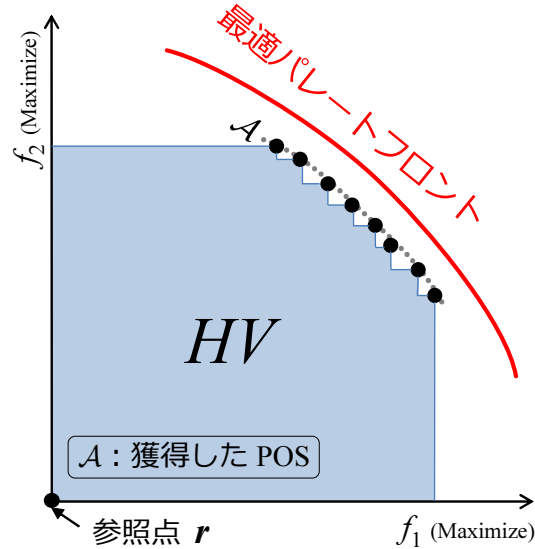


図 4.5: Hypervolume の概念図

とである。パレートフロントの近似精度を高めるためには、以下の 4 つの要素が必要となる。

- (1) 収束性: パレートフロントに対する獲得した POS の収束性の高さ
- (2) 多様性: パレートフロント全体を近似するための獲得した POS の分布の広さ
- (3) 解の数: パレートフロントを精緻に近似するための獲得した POS の解の多さ
- (4) 一様さ: パレートフロントを均一に近似するための獲得した POS の分布の一様さ

上記の 4 要素について順に説明する。まず、(1) 収束性は、獲得した POS のパレートフロントへの近さの観点での評価である。図 4.4(b) に示す二つの解集合では、 $A$  が  $B$  よりパレートフロントへの収束性が高く、良い解集合と判断する。次に、(2) 多様性は、パレートフロント全体を近似するための獲得した POS の分布の広さの観点の評価である。図 4.4(c) に示す二つの解集合では、 $A$  が  $B$  より広範に分布しており、多様性が高く、良い解集合と判断する。(3) 解の数は、パレートフロントを高粒度に近似するための獲得した POS に含まれる解の多さの観点の評価である。図 4.4(d) に示す二つの解集合では、 $A$  が  $B$  よりパレートフロントを近似する解が多いため近似粒度が高く、良い解集合と判断する。最後に、(4) 一様さは、パレートフロントを均一に近似するための獲得した POS の分布の観点の評価である。図 4.4(e) に示す二つの解集合では、 $A$  が  $B$  より均一に分布しておりパレートフロントの近似に斑が少なく、良い解集合と判断する。

これら 4 要素を高めた POS が獲得できるアルゴリズムほど、解探索性能の高いアルゴリズムと判断される。以下、獲得した POS の評価尺度について説明する。

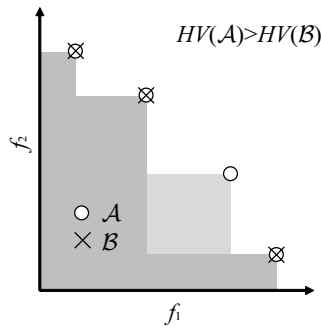


図 4.6: 弱優位の例

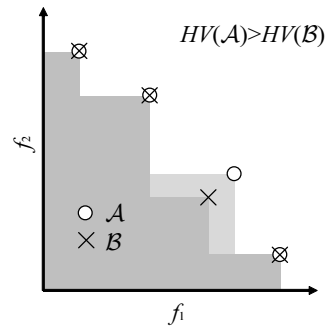


図 4.7: 強優位の例

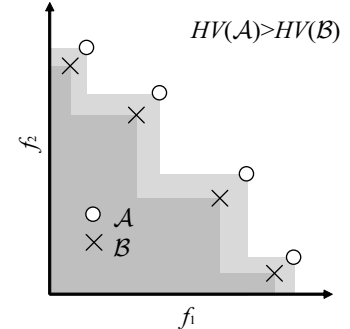


図 4.8: 完全優位の例

#### 4.4.1 Hypervolume

獲得した POS の総合評価尺度として Hypervolume ( $HV$ ) [42] がある.  $m = 2$  目的問題における  $HV$  の概念図を図 4.5 に示す.  $HV$  は, 目的関数空間において獲得した POS と参照点  $r$  で構成される  $m$  次元体積である.  $HV$  の値には, 獲得した POS の最適なパレートフロントに対する収束性の高さ, パレートフロントを広域に近似する多様性の高さ, パレートフロントを精緻に近似するための POS の数の多さ, パレートフロントを均一に近似する POS の分布の一様さの 4.4 節に示した 4 要素すべてが反映される. そのため,  $HV$  が高いほど, 獲得した POS の最適なパレートフロントに対する収束性, 多様性, 解の数, 分布の一様さの点で良好な結果と判断する. 本研究では, 比較するいずれのアルゴリズムも, 獲得した POS を内包する解集団サイズを等しく設定して実験する. すなわち, いずれのアルゴリズムも, 最大で解集団サイズと等しい数の POS を獲得可能という公平な条件下で実験する.

これまでに多数の評価尺度が提案されてきたが, 近年, 最も頻繁に使用される評価尺度が  $HV$  である. その背景には, 文献 [43] による評価尺度の比較の結果,  $HV$  の利用が推奨されたことにある. 二つの非劣解集合を比較するとき, 3 段階の優位性が定義されており [43], 非劣解集合の評価尺度は, その 3 段階の優位性を維持できることが望ましい. 二つの非劣解集合を  $A$  と  $B$  とし,  $ND(S)$  を  $S$  の非劣解集合とすると, 3 段階の優位性は, 以下で定義される.

**弱優位 (Weak outperformance):**  $ND(A \cup B) = A$  かつ  $A \neq B$  のとき, すなわち  $B$  のすべての点が  $A$  の点によって被覆 (同一点かパレート支配) されており, かつ,  $A$  の少なくとも一つの点が  $B$  に含まれないとき,  $A$  は  $B$  より弱優位であるという (図 4.6)

**強優位 (Strong outperformance):**  $ND(A \cup B) = A$  かつ  $B \setminus ND(A \cup B) \neq \emptyset$  のとき,

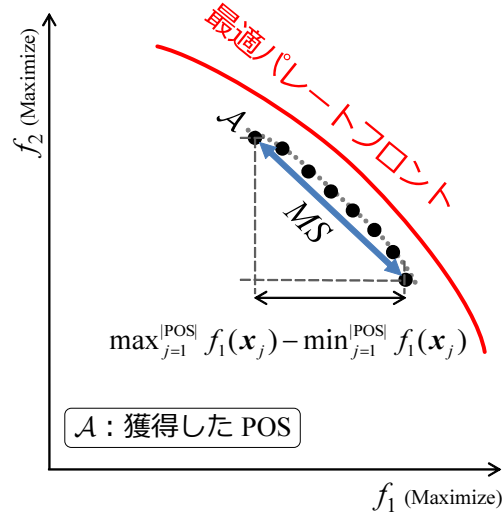


図 4.9: Maximum Spread の概念図

すなわち  $B$  のすべての点が  $A$  の点によって被覆されており、 $B$  のいくつかの点が  $A$  のある点によって支配されるとき、 $A$  は  $B$  より強優位であるという (図 4.7)

**完全優位 (Complete outperformance):**  $ND(A \cup B) = A$  かつ  $B \cap ND(A \cup B) = \emptyset$  のとき、すなわち  $B$  のそれぞれの点が  $A$  のある点に支配されるとき、 $A$  は  $B$  より完全優位であるという (図 4.8)

解集合  $A$  の  $B$  に対する優位性は、弱優位、強優位、完全優位の順に高まる。HV による非劣解集合の順序関係は、これら 3 段階の優位性を維持可能であることが、HV の利用が推奨され、頻繁に利用される主な理由である。他の利点としては、真のパレートフロントが未知の場合に利用可能な点、スカラー値による絶対値評価のためアルゴリズムの順序関係ができること、性能差がスカラー値によって精緻に表現できることがあげられる。これまで、10 目的ほどの最適化の評価尺度として利用されてきており、本論文では、2~6 目的問題までの最適化性能の主たる評価尺度として HV を用いる。

#### 4.4.2 Maximum Spread

獲得した POS の目的空間における広がり測る評価尺度として、式 (4.8) で定義される Maximum Spread (MS) [42] を用いる。  $m = 2$  目的問題における MS の概念図を図 4.9 に示す。MS は値が高いほど、広範なパレートフロントを近似可能な多様性の高い POS と判断する。

$$MS = \sqrt{\sum_{i=1}^m \left( \max_{j=1}^{|POS|} f_i(x_j) - \min_{j=1}^{|POS|} f_i(x_j) \right)^2} \quad (4.8)$$



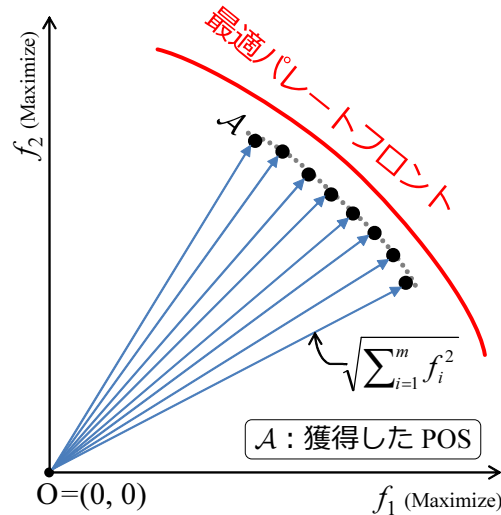


図 4.10: Norm の概念図

#### 4.4.3 C メトリック

獲得した POS の最適なパレートフロントに対する収束性を計測する評価尺度として、次式で算出される C メトリック [43] を用いる。

$$C(\mathcal{A}, \mathcal{B}) = \frac{|\{b \in \mathcal{B} \mid \exists a \in \mathcal{A} : a \succ_f b\}|}{|\mathcal{B}|} \quad (4.9)$$

C メトリックは、二つの解集団  $\mathcal{A}$  と解集団  $\mathcal{B}$  を相対的に評価する方法であり、解集団  $\mathcal{A}$  に支配されている解集団  $\mathcal{B}$  に属する個体の割合を表している。すなわち、 $C(\mathcal{A}, \mathcal{B})$  の値が 1.0 に近いほど解集団  $\mathcal{A}$  は解集団  $\mathcal{B}$  を多く支配するため、 $\mathcal{A}$  が  $\mathcal{B}$  より収束性が高いと判断する。逆に、 $C(\mathcal{A}, \mathcal{B})$  が 0.0 に近いほど、解集団  $\mathcal{A}$  は解集団  $\mathcal{B}$  を支配できていないことを示す。一般に  $C(\mathcal{A}, \mathcal{B}) \neq C(\mathcal{B}, \mathcal{A})$  であり、両者を計測することが重要である。C メトリックでは、パレート支配の概念を用いているため、パレートの形状に依存しない利点がある。ただし、問題の目的数の増加に伴って、解の支配関係に優劣が付きづらくなるため、差がみられにくくなる場合がある。

#### 4.4.4 Norm

獲得した POS の真のパレートフロントへの収束性の評価尺度として、Norm[44] がある。目的数  $m = 2$  の場合の Norm の概念図を図 4.10 に示す。Norm は、以下の式を用いて解集団における評価値ベクトルのノルムの大きさの平均値を計測する。Norm 値が高

いほど，最適なパレートフロントへの収束性が高い POS と判断する．

$$Norm = \frac{\sum_{j=1}^{|POS|} \sqrt{\sum_{i=1}^m f_i(\mathbf{x}_j)^2}}{|POS|} \quad (4.10)$$

*Norm* 値は，パレートフロントの形状や獲得された POS の分布により，その特徴を十分に反映した詳細な収束性の比較が出来ない場合があるが，本論文では，POS の収束性に関する一般的な傾向を観察するために用いている．

## 第 5 章

# 二段階の非支配ソートと指向性交配を用いる MOEA

### 5.1 TNSDM アルゴリズム

本章では，CMOP における MOEA の解探索性能を改善することを目的として，制約違反量と目的関数値による二段階の非支配ソートと探索方向への解の収束を促す指向性交配を導入する TNSDM (Two-stage Non-dominated Sorting and Directed Mating) アルゴリズムを提案する．TNSDM のブロック図を図 5.1 に，擬似コードを **Algorithm 1** に示す．また，擬似コードに使用される記号の説明を表 5.1 に示す．TNSDM は NSGA-II と同様に，エリートである親集団  $\mathcal{P}$  と子集団  $\mathcal{Q}$  から構成される全解集団  $\mathcal{R} (\mathcal{P} \cup \mathcal{Q})$  を用いるモデルを基に設計している．本章では，親集団  $\mathcal{P}$  と交配ペアの選択法を新たに提案し，交叉と突然変異法は既存の方法を利用する．TNSDM では，新たなパラメータを追加しない．

#### 5.1.1 二段階の非支配ソート

TNSDM は，全解集団  $\mathcal{R}$  から親集団  $\mathcal{P}$  を選択するために，全解集団  $\mathcal{R}$  を二段階の非支配ソートによってフロント分類する (**Algorithm 1** の 3–8 行目)．TNSDM は，制約違反量に基づくパレート支配 [21, 38, 39] を用いる．解  $\mathbf{x}$ ,  $\mathbf{y}$  が，次式を満たすとき  $\mathbf{x}$  は  $\mathbf{y}$  を制約違反量において支配する ( $\mathbf{x} \succ_v \mathbf{y}$ ) という．

$$\forall j : v_j(\mathbf{x}) \leq v_j(\mathbf{y}) \quad \wedge \quad \exists j : v_j(\mathbf{x}) < v_j(\mathbf{y}) \\ (j = 1, 2, \dots, k) \tag{5.1}$$

TNSDM では，まず，式 (5.1) を用いて，全解集団  $\mathcal{R}$  を制約違反量において支配されない順に，非支配ソートによって，複数のフロント ( $\mathcal{F}_1^v, \mathcal{F}_2^v \dots$ ) へ分類する．実行可能解

## 二段階の非支配ソーティング

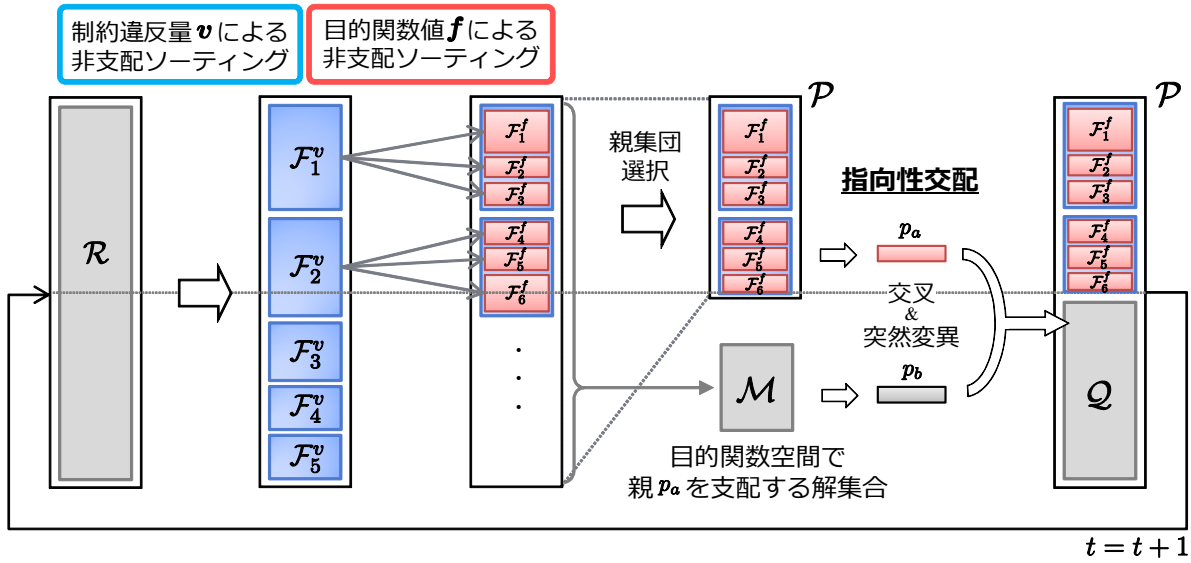


図 5.1: TNSDM のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団)

は、 $\mathbf{v}(\mathbf{x}) = (0, 0, \dots, 0)$  であるため、必ず最上位フロント  $\mathcal{F}_1^v$  に分類される。次に、それぞれのフロント  $\mathcal{F}_i^v$  ( $i = 1, 2, \dots$ ) に対して、式 (2.2) を用いて目的関数値において支配されない順に、非支配ソートによって、さらに細分化したフロント ( $\mathcal{F}_1^f, \mathcal{F}_2^f, \dots$ ) へ分類する。以降は、従来の CNSGA-II と同様に、図 5.1 の水平の破線で示す全解集団  $\mathcal{R}$  の半数を、次世代の親集団  $\mathcal{P}$  として選択する。具体的には、 $\mathcal{F}_1^f, \mathcal{F}_2^f, \dots$  のフロント番号が小さい上位フロントから順に、親集団  $\mathcal{P}$  へ加えていく。図 5.1 の  $\mathcal{F}_6^f$  のように、あるフロントに属するすべての解を親集団  $\mathcal{P}$  に加えると、必要とする親集団サイズを超えてしまう場合、そのフロントに属する解について、目的関数空間における混雑度距離 CD [14] が長い解から順に、親集団サイズを満たすまで  $\mathcal{P}$  に加える。

これにより、制約違反量によるフロントが同一の解の優劣を、目的関数値によるフロントで決定し、目的関数値の良好な実行可能解を獲得しやすくする。

### 5.1.2 解の探索方向を考慮した指向性交配

TNSDM では、探索方向への解の収束を促す交配法を導入する。概念図を図 5.2 に示す。この例では、解集団  $\mathcal{R}$  が、 $\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_6^f$  に分類されており、実行可能解である  $\mathcal{F}_1^f$  と  $\mathcal{F}_2^f$  が親集団  $\mathcal{P}$  になる。従来の CNSGA-II の場合、すべての交配は、親集団  $\mathcal{P}$  ( $= \mathcal{F}_1^f \cup \mathcal{F}_2^f$ ) 内で行われる。一方、TNSDM では、片方の親は  $\mathcal{P}$  から選択されるが、もう片方の親は  $\mathcal{P}$  に選出されなかった実行不可能解から選択されることがある。この方法を指向性交配法と称し、詳細を以下と、Algorithm 1 の 11–20 行目に示す。

---

**Algorithm 1** TNSDM

---

```
1: Initialize ( $\mathcal{R}$ )
2: for  $t = 1$  to  $T$  do
3:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
4:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sort } (\mathcal{R}, \succ_v)$ 
5:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
6:      $j = 1 + |\mathcal{F}^f|$ 
7:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sort } (\mathcal{F}_i^v, \succ_f)$ 
8:   end for
9:   Crowding distance ( $\mathcal{F}^f$ )
10:   $\mathcal{P} = \text{Truncation } (\mathcal{R}, |\mathcal{R}|/2)$ 
11:  for  $j = 1$  to  $|\mathcal{Q}|$  do
12:     $p_a = \text{Tournament selection } (\mathcal{P})$ 
13:     $\mathcal{M} = \{\mathbf{x} \in \mathcal{R} \mid \mathbf{x} \succ_f p_a\}$ 
14:    if  $p_a$  is feasible and  $|\mathcal{M}| \geq 2$  then
15:       $p_b = \text{Tournament selection } (\mathcal{M}) : \textit{Directed Mating}$ 
16:    else
17:       $p_b = \text{Tournament selection } (\mathcal{P}) : \textit{Conventional Mating}$ 
18:    end if
19:     $Q_j = \text{Crossover and Mutation } (p_a, p_b)$ 
20:  end for
21:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
22: end for
23:  $POS = \{\mathbf{x} \in \mathcal{R}^{feasible} \mid \neg \exists \mathbf{y} \in \mathcal{R}^{feasible} : \mathbf{y} \succ_f \mathbf{x}\}$ 
```

---

ひとつの子を生成するために、まず、親集団  $\mathcal{P}$  からトーナメントによって親  $p_a$  を選択する。このトーナメントでは、 $\mathcal{P}$  からランダムに選択した2個体のうち、フロント番号が小さい上位フロントの解を選択し、同一フロントの場合には、混雑度距離  $CD$  が長い解を選択する。次に、実行不可能解を含むすべての解集団  $\mathcal{R}$  から、親  $p_a$  を目的関数値において支配する解集合  $\mathcal{M} (= \{\mathbf{x} \in \mathcal{R} \mid \mathbf{x} \succ_f p_a\})$  を選出する。 $p_a$  が実行可能解でかつ、 $|\mathcal{M}| \geq 2$  の場合、 $\mathcal{M}$  からトーナメント選択によって親  $p_b$  を選択する。それ以外の場合、親集団  $\mathcal{P}$  からトーナメント選択によって親  $p_b$  を選択する。ここで、指向性交配を行う条件として解集合  $\mathcal{M}$  に含まれる解の個数を2以上とする理由は、トーナメント選択を行って良い解を親  $p_b$  として選択するためである。

図 5.2 の例では、親  $p_a$  を支配する3つの実行不可能解が  $\mathcal{M}$  になる。 $\mathcal{M}$  からランダム

表 5.1: 擬似コード内で使用される記号 (TNSDM)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $Q = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

に取り出された  $\mathcal{F}_5^f$  と  $\mathcal{F}_3^f$  に属する解がトーナメント選択によって比較され, フロント番号が小さい  $\mathcal{F}_3^f$  に属する解が, もう片方の親  $p_b$  になる.  $p_b$  は実行不可能解だが,  $p_a$  を目的空間で支配するため,  $p_a$  を最適なパレートフロントの方向にさらに進化させるために, 有効な遺伝子を有する可能性があり, これらの交配によって解探索性能の改善を狙う.

## 5.2 実験内容

提案する TNSDM の解探索性能を, 従来法の CNSGA-II, RTS-O, RTS-C と比較する. 本章では, 連続値の変数空間を有するテスト問題として, SRN, TNK, OSY を用いる. また, 離散値の変数空間を有するテスト問題として,  $mk$ -KP を用いる. さらに, 実問題の 3 段式ハイブリッドロケットの概念設計問題を用いる. また, TNSDM と CNSGA-II の計算時間や  $mk$ -KP における動的計画法 [45] との比較, また, 真の最適解との比較を行う.

### 5.2.1 パラメータ

本章では, すべてのアルゴリズムで共通のパラメータを用いる. 解集団サイズは,  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) とした. 一般的に解集団サイズ  $N$  が大きいほど, パレートフロントの近似精度は高まる. ここで, サイズ  $N = 200$  の解集団を用いることは, 最大  $N$  点を用いてパレートフロント全体の近似を試みることを意味する. 連続問題と 3 段

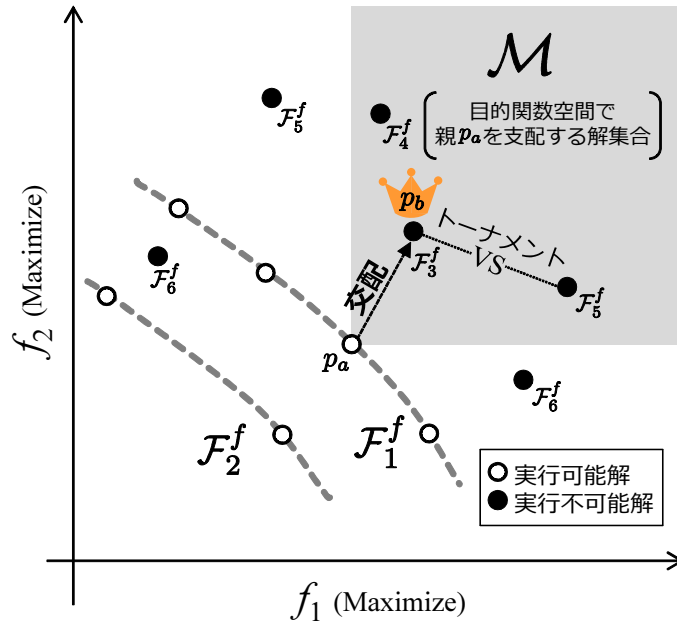


図 5.2: 親  $p_a$  の探索方向を考慮した指向性交配 (最大化問題の例)

式ハイブリッドロケットの概念設計問題において、一般的に用いられる遺伝的パラメータとして、交叉には SBX ( $\eta_c = 15$ , 交叉率  $P_c = 0.8$ ), 突然変異には Polynomial mutation ( $\eta_m = 20$ , 突然変異率  $P_m = 0.2$ ) を用いた. 探索終了世代数はそれぞれ, 連続問題で世代数  $T = 1,000$  世代, 3 段式ハイブリッドロケットの概念設計問題で世代数  $T = 200$  世代とした.  $mk$ -KP のパラメータは, アイテム数  $n = 500$ , 目的数  $m = 2$ , 制約数  $k = \{2, 4, 6, 8, 10\}$ , 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  とし, 文献 [26] に倣って価値と重さは  $[10, 100]$  の範囲のランダムな整数値に設定した.  $mk$ -KP における遺伝的パラメータとして, 世代数  $T = 10,000$  世代, 一般的に用いられる交叉法の一様交叉 (交叉率  $P_c = 1.0$ ), 突然変異にはビット反転法 (変異率  $P_m = 1/n$ ) を用いた. 結果は, 各アルゴリズムの 100 回の試行の平均値を用いて比較する.

## 5.2.2 評価尺度

獲得した解集合の総合評価尺度として Hypervolume ( $HV$ ) を用いる.  $HV$  を計測するための参照点は, SRN の場合  $\mathbf{r} = (250, 0)$ , TNK の場合  $\mathbf{r} = (1.2, 1.2)$ , OSY の場合  $\mathbf{r} = (-30, 80)$ ,  $mk$ -KP の場合  $\mathbf{r} = (0, 0)$  とした. 3 段式ハイブリッドロケットの概念設計問題では各目的関数のとる値の範囲が異なるため, 世代を通じて得られた各目的関数の最大, 最小値で正規化された目的関数値と  $\mathbf{r} = (1.0, 0.0)$  を用いて  $HV$  を算出する. また, それぞれ多様性の評価に Maximum Spread ( $MS$ ), 収束性の評価に  $C$  メトリックを用

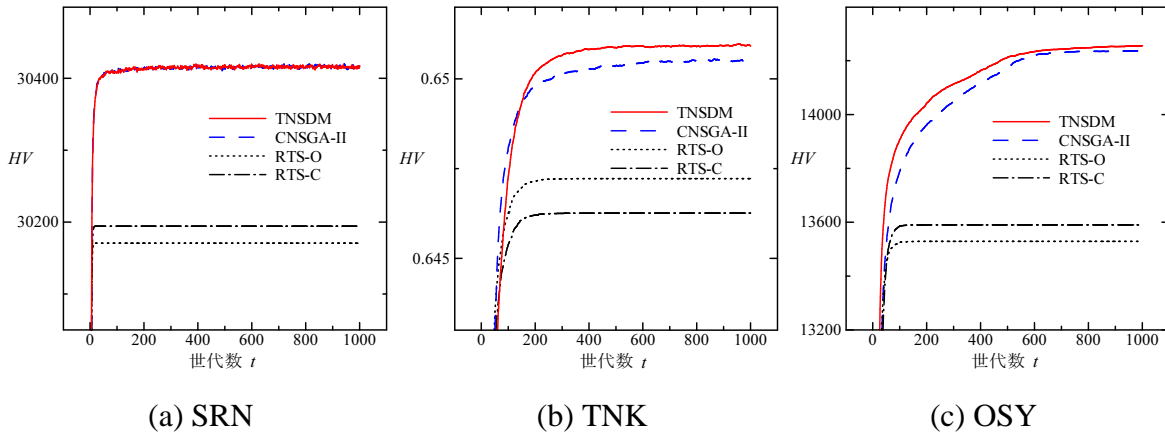


図 5.3: 連続問題における  $HV$  の世代推移 (総合評価)

いる。

## 5.3 実験結果と考察

### 5.3.1 連続問題における $HV$ の比較

まず、連続問題の SRN, TNK, OSY における TNSDM, 従来法の CNSGA-II, RTS-O, RTS-C による  $HV$  の世代推移を図 5.3 に示す。

図 5.3 (a) から、SRN において、TNSDM は従来法の CNSGA-II と同等の  $HV$  を示すことがわかる。一方、図 5.3 (b) と (c) から、TNK と OSY において、TNSDM は CNSGA-II より高い  $HV$  を示すことがわかる。TNK において、従来法の CNSGA-II による  $HV$  は、約 200 世代ほどで収束しており、その後、世代数を費やしても TNSDM による  $HV$  に達することはない。また、RTS-O と RTS-C は、すべての問題において、TNSDM と CNSGA-II より低い  $HV$  を示すことがわかる。これらの結果から、TNSDM は、比較したすべての連続問題において、最も  $HV$  の高い解探索性能を示すことが確かめられた。従来法の中で最も高い  $HV$  を示す CNSGA-II と提案する TNSDM を比較すると、POS が変数空間の実行可能領域の内部に存在する SRN において、 $HV$  の差は見られない。一方、POS が変数空間の実行可能領域と実行不可能領域の境界に存在する TNK と OSY において、TNSDM の方が高い  $HV$  を示すことがわかる。これは、POS が実行可能領域と実行不可能領域の境界に存在するため、その境界を探索するために、指向性交配における第二の実行不可能解の親が解探索を促進する効果をもたらした結果と考えられる。また、非連続のパレートフロントを有する TNK と凸型のパレートフロントを有する OSY の両方において TNSDM が良好な結果を示したことは、パレートフロントの形状に対する解探索性能の頑健性の高さを示している。



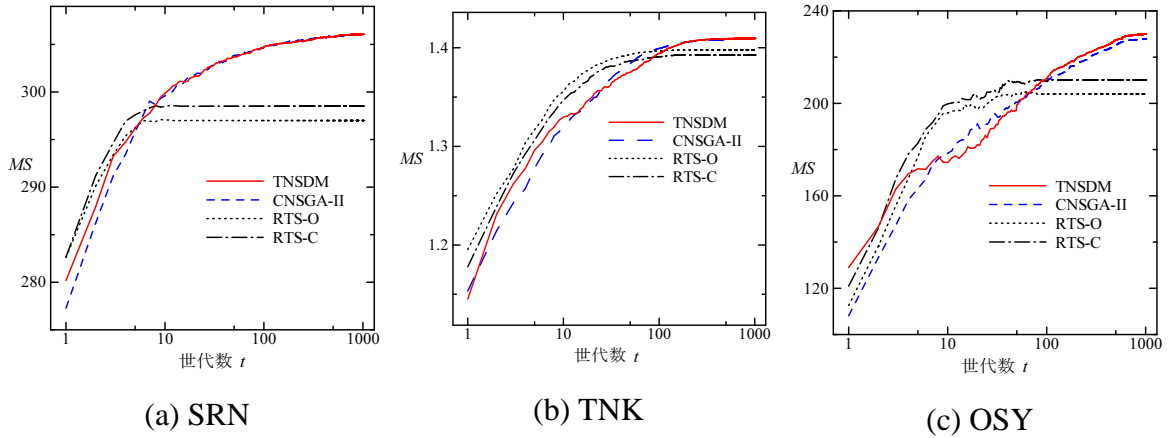


図 5.4: 連続問題における  $MS$  の世代推移 (多様性評価)

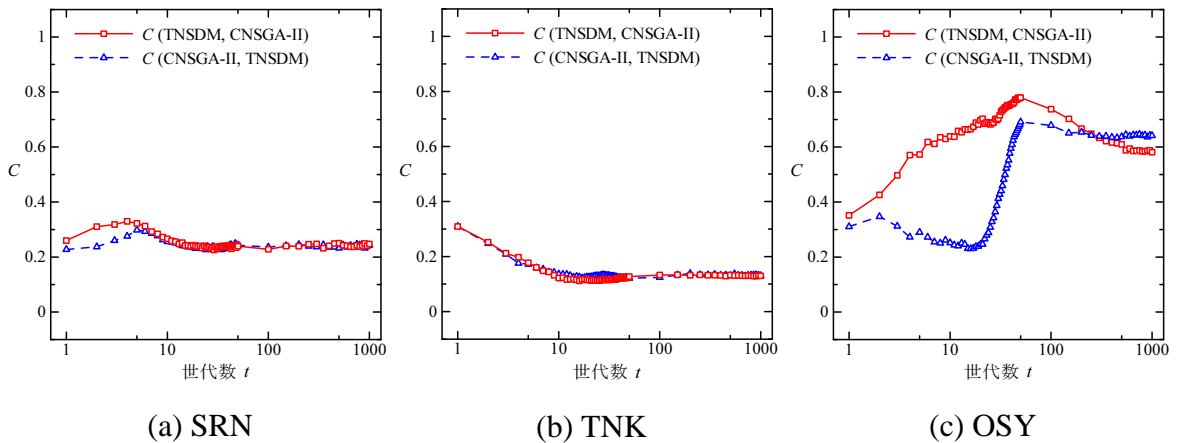


図 5.5: 連続問題における  $C$  メトリック比較の世代推移 (収束性評価)

### 5.3.2 連続問題における POS の多様性と収束性

TNSDM が、連続問題において高い  $HV$  を達成した要因を調査するために、各アルゴリズムによって獲得した POS を、目的関数空間における多様性と、収束性にかけて分析する。獲得した POS の多様性を比較するために、 $MS$  の世代推移を図 5.4 に示す。また、獲得した POS の収束性を比較するために  $C$  メトリックの結果を図 5.5 に示す。 $C$  メトリックは、POS を絶対評価する評価尺度ではなく、二種類の POS を相対比較する評価尺度である。ここでは、図 5.3 で高い  $HV$  を示した TNSDM と CNSGA-II による各世代の POS を比較した結果のみを掲載する。特に解探索初期に表れる特徴を観察するために、図 5.4 と図 5.5 では、横軸を対数スケールにしていることに注意されたい。

まず、SRN について、TNSDM と CNSGA-II を比較する。図 5.4 (a) から、探索初期で

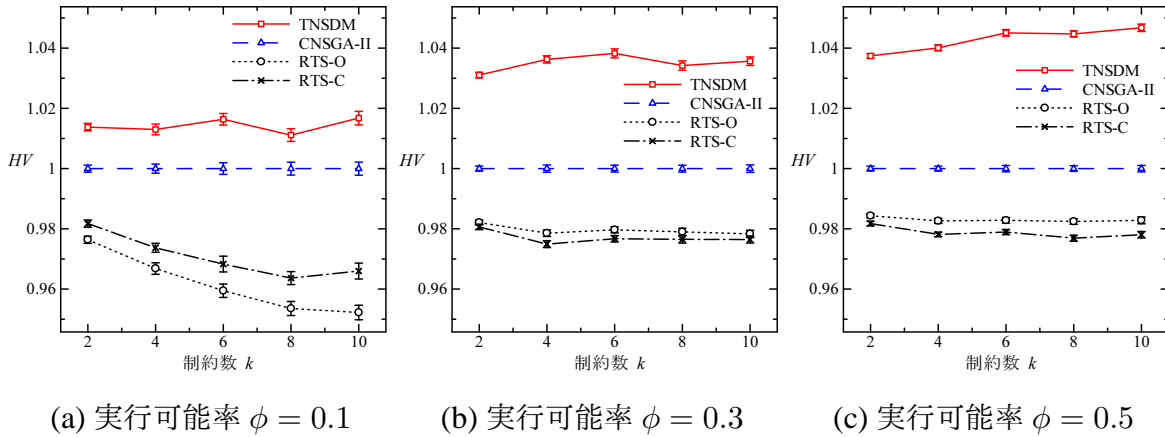


図 5.6:  $mk$ -KP における最終世代の  $HV$  (総合評価)

は、TNSDM が CNSGA-II より高い  $MS$  を示すことがわかる。しかし、最終世代では、TNSDM と CNSGA-II の  $MS$  が同等になり、POS の多様性に差がなくなる。また、図 5.5 (a) から、探索初期では、 $C(\text{TNSDM}, \text{CNSGA-II})$  が  $C(\text{CNSGA-II}, \text{TNSDM})$  より高い値を示すことがわかる。これは、TNSDM の POS が、従来の CNSGA-II の POS をより多く支配しており、収束性が高いことを示している。しかし、解探索の進行に伴って、その差が無くなることがわかる。このように、SRN においては、約 10 世代までは、TNSDM は CNSGA-II より多様性と収束性の両面で優れた POS を獲得するが、それ以降は、多様性と収束性に差がつかないため、図 5.3 (a) で  $HV$  に差が表われないと考えられる。

次に、TNK について、TNSDM と CNSGA-II を比較する。図 5.4 (b) から、最終世代において、TNSDM が CNSGA-II は、ほぼ同等の  $MS$  を示している。また、図 5.5 (b) から、TNSDM と CNSGA-II は、ほぼ同等の収束性を示していることがわかる。この結果から、TNSDM は得られる POS の解の数や一様さの観点で優れており、図 5.3 (b) において、CNSGA-II より高い  $HV$  を達成したと考えられる。

最後に、OSY について図 5.5 (c) から、探索初期の TNSDM の POS は、CNSGA-II より高い収束性を実現することがわかる。しかし、約 100 世代を過ぎると、 $C(\text{TNSDM}, \text{CNSGA-II})$  が  $C(\text{CNSGA-II}, \text{TNSDM})$  より小さくなる傾向が見て取れる。これは、図 5.4 (c) において、TNSDM の  $MS$  が、CNSGA-II より高くなる世代数とほぼ一致している。すなわち、TNSDM は探索終盤に向けて多様性の高い POS を獲得するようになる。C メトリックは、支配する解の割合を計測するため、TNSDM が目的空間を広域に探索すると、目的空間の一部に偏って分布する CNSGA-II の POS を支配しにくくなり、 $C(\text{TNSDM}, \text{CNSGA-II})$  が低下したと考えられる。

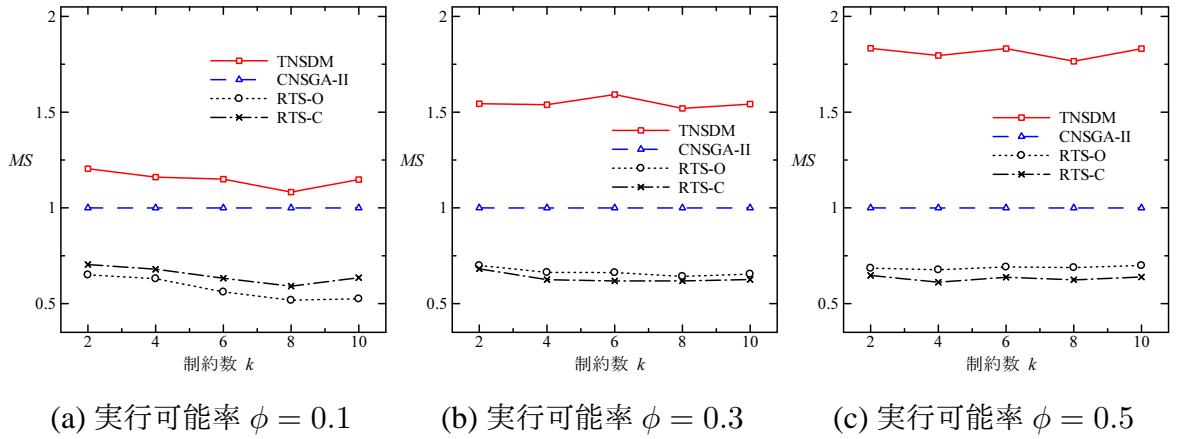


図 5.7:  $mk$ -KP における最終世代の  $MS$  (多様性評価)

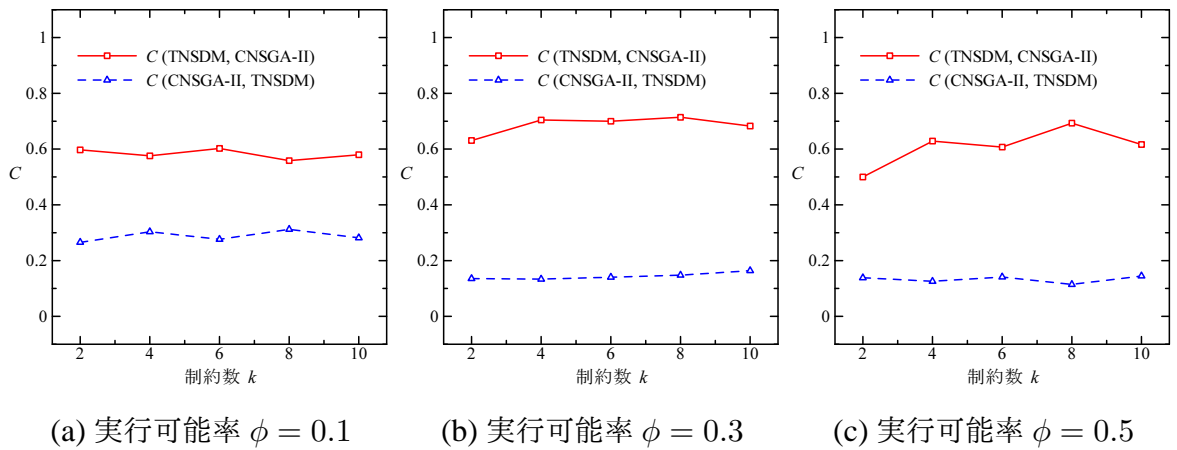


図 5.8:  $mk$ -KP における  $C$  メトリックによる比較 (収束性評価)

### 5.3.3 $mk$ -KP における $HV$ の比較

次に、目的数  $m = 2$ 、制約数  $k = \{2, 4, 6, 8, 10\}$ 、実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の  $mk$ -KP における最終世代の  $HV$  を図 5.6 に示す。ここで、すべての結果は、従来の CNSGA-II による  $HV$  を 1.0 として正規化してプロットしている。また、エラーバーは 95% 信頼区間を示している。

この結果から、TNSDM は、制約数  $k$  や実行可能率  $\phi$  によらず、従来の CNSGA-II より高い  $HV$  を達成することがわかる。また、従来の RTS-O と RTS-C を比較すると、実行可能率が最も低い  $\phi = 0.1$  のときのみ、制約違反量の非支配ランク  $R_{con}$  の小さい親が選ばれやすい RTS-C が、RTS-O より高い  $HV$  を示していることがわかる。

このように図 5.3 と図 5.6 の結果から、本章で使用したすべてのテスト問題において、

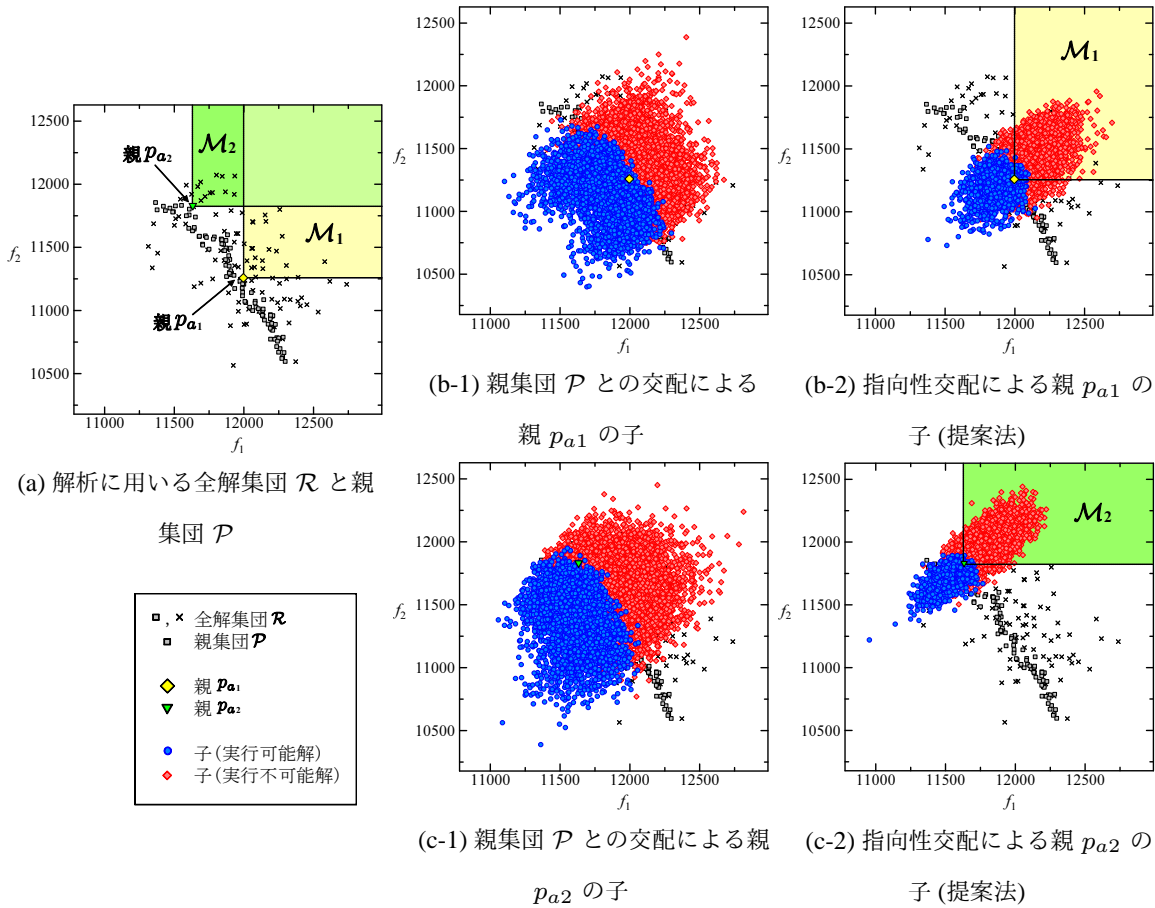


図 5.9: 指向性交配によって生成された子の目的関数空間における分布 ( $mk$ -KP)

TNSDM は最も高い  $HV$  を示すことが確かめられた。

### 5.3.4 $mk$ -KP における POS の多様性と収束性

TNSDM が、 $mk$ -KP において高い  $HV$  を達成した要因を調査するために、各アルゴリズムによって獲得した POS を、目的関数空間における多様性と、収束性にわけて分析する。

制約数  $k$  と実行可能率  $\phi$  を変化させた場合の最終世代の  $MS$  を図 5.7 に示す。図 5.6 と同様に、すべての結果は、従来の CNSGA-II による  $MS$  を 1.0 として正規化してプロットしている。この結果から、すべての実行可能率  $\phi$  と制約数  $k$  において、TNSDM は比較するアルゴリズムの中で、最も高い  $MS$  を示しており、獲得した POS の目的関数空間における多様性が最も高いことがわかる。また、これらの  $MS$  の結果は、図 5.6 に示した  $HV$  の傾向に似ていることがわかる。

次に、獲得した POS の収束性を  $C$  メトリックを用いて比較する。ここでは、図 5.5 と

同様に、TNSDM と CNSGA-II のみを取り上げて比較する。制約数  $k$  と実行可能率  $\phi$  を変化させた場合の最終世代の  $C$  メトリックの結果を図 5.8 に示す。この結果から、すべての問題について、 $C(\text{CNSGA-II}, \text{TNSDM})$  より、 $C(\text{TNSDM}, \text{CNSGA-II})$  のほうが高い値を示すことがわかる。これは、TNSDM によって獲得した POS が、CNSGA-II によって獲得した POS をより多く支配していることを示している。すなわち、TNSDM によって獲得した POS の方が、CNSGA-II より最適なパレートフロントに対する収束性が高いといえる。

これらの  $MS$  と  $C$  メトリックの結果から、 $mk$ -KP において、TNSDM は多様性と収束性の両面で、従来法より優れた POS を獲得できるため、図 5.6 で最も高い  $HV$  を示したといえる。

### 5.3.5 指向性交配による子の分布

指向性交配によって生成される子の目的関数空間における分布について解析する。ここでは、 $m = 2$  目的、 $k = 6$  制約、実行可能率  $\phi = 0.3$  の  $mk$ -KP を用い、図 5.9 (a) に示す、TNSDM によって進化させた  $t = 300$  世代目の全解集団  $\mathcal{R}$  と親集団  $\mathcal{P}$  を解析に用いる。図中の二つの親  $p_{a1}$  と  $p_{a2}$  に着目し、もう片方の親を、指向性交配によって選択した場合の子と、CNSGA-II のように親集団  $\mathcal{P}$  から選択した場合の子を、擬似的に 10,000 個体ずつ生成する。指向性交配において、親  $p_{a1}$  の交配候補になる解の存在領域を  $\mathcal{M}_1$ 、親  $p_{a2}$  の交配候補になる解の存在領域を  $\mathcal{M}_2$  として図 5.9 (a) に図示した。  $\mathcal{M}_1$  と  $\mathcal{M}_2$  共に、親集団  $\mathcal{P}$  に選択されない解が交配候補になることがわかる。

親  $p_{a1}$  の交配相手を、親集団  $\mathcal{P}$  から選択した場合の子の分布を図 5.9 (b-1) に示す。この場合、パレートフロントを近似するために解集団内に維持されている目的関数値と遺伝子構造の多様な親集団  $\mathcal{P}$  から、もう片方の親を選択するため、生成された子が目的関数空間の広域に分布することがわかる。次に、親  $p_{a1}$  の交配相手を、指向性交配によって  $\mathcal{M}_1$  から選択した場合の子の分布を図 5.9 (b-2) に示す。この場合、親  $p_{a1}$  の周辺に子が集中的に生成されることがわかる。また、図 5.9 (b-1) と比較して、指向性交配の方が  $p_{a1}$  を支配する実行可能解の子が多いことがわかる。

同様に、親  $p_{a2}$  の交配相手を、親集団  $\mathcal{P}$  から選択した場合の子の分布を図 5.9 (c-1) に、指向性交配によって  $\mathcal{M}_2$  から選択した場合の子の分布を図 5.9 (c-2) に示す。図 5.9 (c-1) から、図 5.9 (b-1) と同様に、親集団  $\mathcal{P}$  内で交配すると、子が目的関数空間の広範囲に分布することがわかる。また、この場合、目的関数値の著しく低い子が多数生成されることがわかる。一方、図 5.9 (b-2) と図 5.9 (c-2) から、指向性交配を用いると、片方の親の存在位置を中心に、さらに目的関数値を高める方向に子が集中的に生成されることがわかる。このように、指向性交配は、親集団  $\mathcal{P}$  に選択されない解を交配相手として活用することによって、片方の親の探索方向に、集中的に子を生じることが確かめられた。

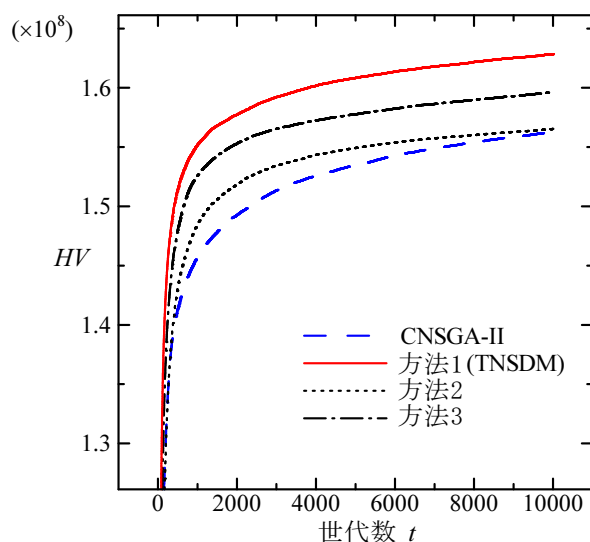


図 5.10: 二段階ソートによる解のランキングの効果 ( $mk$ -KP)

指向性交配では、親集団  $\mathcal{P}$  から選択した片方の親  $p_a$  より高い目的関数値を持つ子が生成されることを期待している。しかし、変数空間と目的関数空間の関係は基本的に不明なため、そのような子を意図的に生成することは困難である。図 5.9 の (b-2) と (c-2) においては、実験で使用した一様交叉とビット反転突然変異によって生成された子が、 $mk$ -KP の目的関数空間において、2 つの親の周辺に高確率で分布するため、指向性交配の効果が得られると考えられる。目的関数空間で近い親は、変数空間でも近く、生成される子も、目的関数空間と変数空間の双方において親の周辺に高確率で分布するという前提のもとに、指向性交配が効果をもたらしているといえる。そのため、変数空間と目的関数空間における解の分布の相関が低い問題においては、図 5.9 の (b-2) と (c-2) のように、2 つの親の近くに子が生成されにくい可能性がある。

仮に、最適化する問題に対する事前知識に基づいて、意図的に 2 つの親の付近に子を生成可能ならば、より効率的な解探索を実現できる可能性がある。しかし、事前知識を用いずとも、変数空間と目的関数空間における解の分布に相関がある問題であれば、図 5.9 の (b-2) と (c-2) のように、従来の一様交叉やビット反転突然変異法によって指向性交配の効果が得られると考えられる。

### 5.3.6 二段階の非支配ソートによる解のランキングの効果

最後に、二段階の非支配ソートによって得られたフロントによる解の優劣を、指向性交配におけるトーナメントの選択基準に利用することの効果について検証する。ここでは、交配する候補解集合  $\mathcal{M}$  からランダムに選択した 2 つの解のうち、フロント番号が小さい上位フロントの解を親  $p_b$  とする方法 1 (TNSDM) と、逆にフロント番号が大きい下位フ

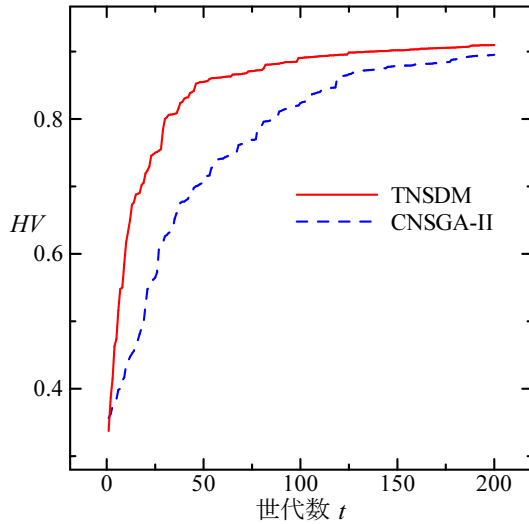


図 5.11: 3 段式ハイブリッドロケットの概念設計問題における  $HV$  の世代推移

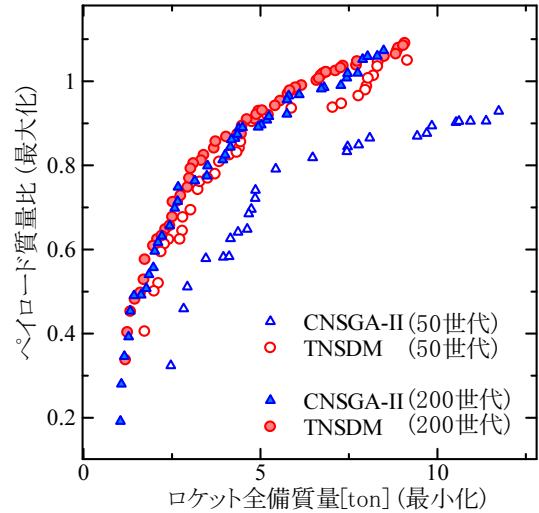


図 5.12: 3 段式ハイブリッドロケットの概念設計問題において獲得した解集合

ロントの解を親  $p_b$  とする方法 2, フロント番号によらずランダムに親  $p_b$  を決定する方法 3 の解探索性能を比較する.  $m = 2$  目的,  $k = 6$  制約, 実行可能率  $\phi = 0.3$  の  $mk$ -KP を用い, これら 3 種類の方法と従来法の CNSGA-II の  $HV$  の世代推移を図 5.10 に示す.

この結果から, まず, ランダムに親を選択する方法 3 が CNSGA-II より高い  $HV$  を示すことがわかる. これは, 親  $p_a$  を目的関数値で支配する個体集合  $M$  からもう一方の親  $p_b$  を選択することによって解探索性能が改善することを示している. 次に, 方法 2 は方法 3 より  $HV$  が低く, 方法 1(TNSDM) は方法 3 より  $HV$  が高いことがわかる. これは,  $M$  からもう片方の親  $p_b$  を選択する方法が解探索性能に影響を与えることを示しており, フロント番号が小さく, 制約違反量の小さくかつ目的関数値が良好な個体を親  $p_b$  に選択する方法 1(TNSDM) を採用した場合に最も高い解探索性能を達成することが確かめられた.

## 5.4 3 段式ハイブリッドロケットの概念設計最適化における解探索性能

従来法の CNSGA-II と TNSDM による  $HV$  の世代推移を図 5.11 に示す. これより, 実行不可能解を積極活用する TNSDM は, 解探索初期の  $HV$  の上昇が著しく, 短時間で高精度の POS を獲得できることがわかる. また, TNSDM の  $HV$  は, 全世代を通して CNSGA-II より高いことがわかる. 次に, 獲得した POS を図 5.12 にプロットした. 200 世代まで時間を十分にかけて最適化すると, 図 5.11 で示した  $HV$  の値も近くなるように, TNSDM と CNSGA-II の POS の分布も近くなる. 注目すべきは, 世代数の少ない 50

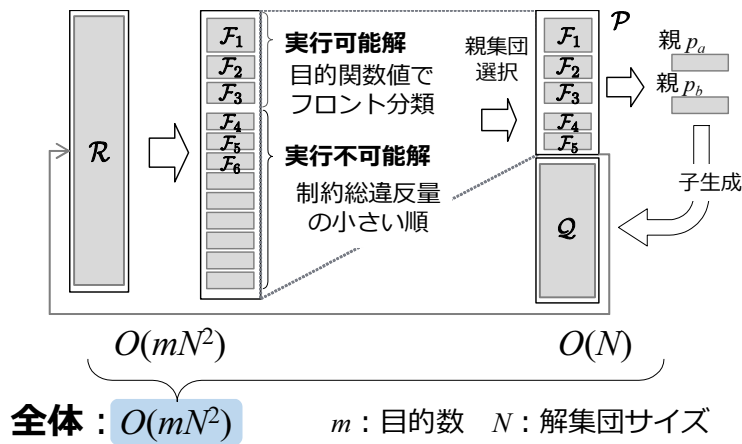


図 5.13: CNSGA-II の計算コスト (ビッグ・オー)

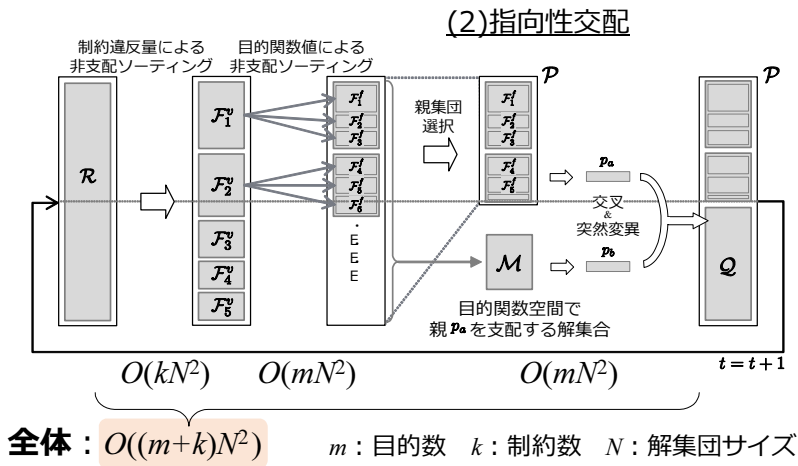


図 5.14: TNSDM の計算コスト (ビッグ・オー)

世代の結果である。50 世代において、実行不可能解を活用する TNSDM による POS は、CNSGA-II の POS をすべて支配するだけでなく、200 世代の POS に近い分布を示すことがわかる。すなわち、TNSDM は、3 段式ハイブリッドロケットの概念設計問題において、解探索中に生成された実行不可能解を活用することによって、短時間で精度の高い POS の獲得に成功したといえる。設計最適化などの実問題では、解一つひとつの評価シミュレーションに膨大な時間を要す場合が多く、最適化の過程で多くの解を生成し評価することが容易ではない。そのため、少ない世代数でより精度の高い解を獲得することは重要である。



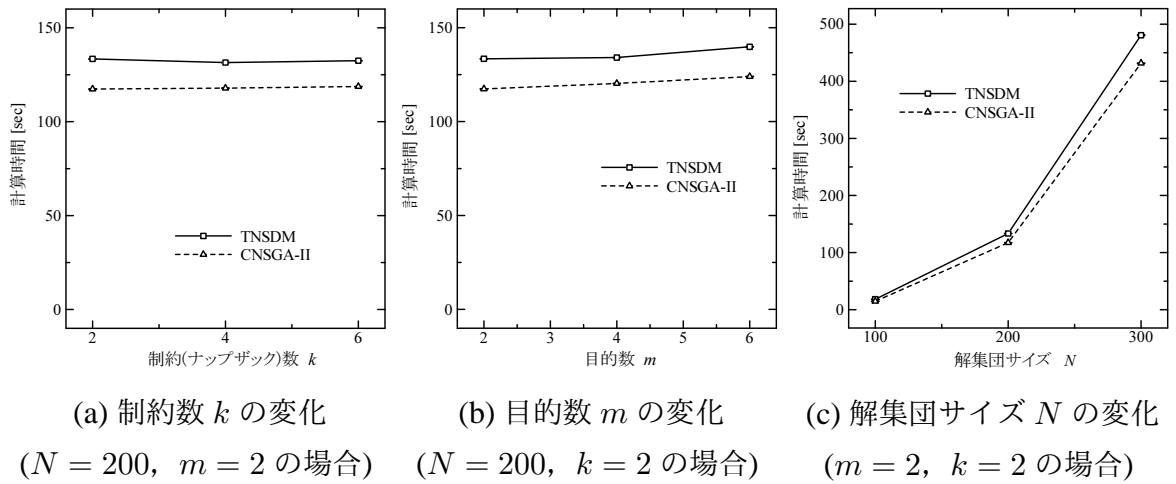


図 5.15:  $mk$ -KP における計算時間の比較

## 5.5 計算時間

次に、TNSDM と従来法の中で最も高い解探索性能を示す CNSGA-II の計算コストを比較する。

まず、計算コストをビッグ・オー記法で比較する。図 5.13 と図 5.14 はそれぞれ CNSGA-II と TNSDM の各処理にかかる計算コストと全体の計算コストをブロック図中に示している。目的数を  $m$ 、解集団サイズを  $N$  としたとき、従来法の CNSGA-II は解のフロント分類に  $O(mN^2)$ 、子集団生成に  $O(N)$  で、全体で  $O(mN^2)$  になる。一方、提案法の TNSDM は、制約数を  $k$  とすると、第一段階の非支配ソートに  $O(kN^2)$ 、第二段階の非支配ソートに  $O(mN^2)$ 、指向性交配による子集団生成に  $O(mN^2)$  の計算コストが必要になり、全体では  $O((m+k)N^2)$  になる。

次に、 $mk$ -KP を用いて、実際の計算時間を比較した結果を図 5.15 に示す。計算機環境として、Intel Core i7-3770 (3.4GHz) の CPU, 8 GB のメインメモリ, OS は Windows 7 を用いた。この結果には、最適化アルゴリズム以外の解の評価や記録処理などのオーバーヘッドが含まれるため、必ずしも上記の計算のオーダーが反映されないことがあることにご注意されたい。まず、制約数  $k$  と目的数  $m$  を変化させた場合の図 5.15 (a) と図 5.15 (b) の結果から、CNSGA-II, TNSDM とともに、制約数  $k$  の増加より目的数  $m$  の増加によって計算時間が長くなる傾向がみられる。次に、解集団サイズ  $N$  を変化させた場合の図 5.15 (c) の結果から、CNSGA-II, TNSDM とともに、解集団サイズ  $N$  の増加によって著しく計算時間が長くなることがわかる。これは、CNSGA-II と TNSDM とともに計算コストが  $N^2$  に従うことによる。図 5.15 (a)- (c) のいずれの問題でも、TNSDM は、CNSGA-II より計算時間がかかることがわかる。これは、TNSDM が、二段階に非支配ソートすることと指

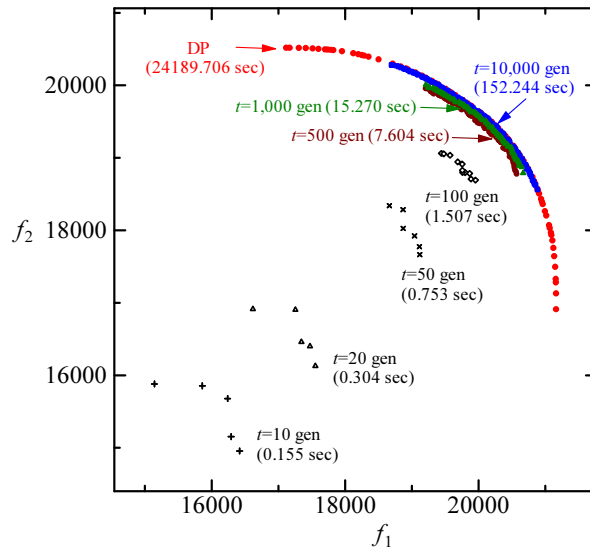


図 5.16:  $mk$ -KP において動的計画法と TNSDM により獲得した解集合

向性交配することによる。TNSDM の計算時間は、CNSGA-II の約 114% だった。

## 5.6 動的計画法との比較

$mk$ -KP( $m = 2$  目的,  $k = 1$  制約, 実行可能率  $\phi = 0.5$ ) を用い, 問題の事前知識を利用して最適化する動的計画法 (Dynamic Programming, 以下 DP) [45] と, 提案法の TNSDM によって獲得した解集合を比較した結果を図 5.16 に示す. DP では, 各目的に重み付けした単一目的最適化を繰り返すソルバーを用いた. 具体的には, 第一目的  $f_1$  の重みを  $i/500$ , 第二目的  $f_2$  の重みを  $(500 - i)/500$  とする加重和による単一目的最適化を  $i = 0, 1, \dots, 500$  まで繰り返す. 計算機環境として, Intel Core i7-3770 (3.4GHz) の CPU, 8 GB のメインメモリ, OS は Windows 7 を用いた実験において, DP による最適解集合の獲得には, 約 6.72 時間を要した. 図 5.16 において, 進化計算による TNSDM は, 複数の世代における解集合をプロットした. 解集合は第 1 試行の結果で, 併記した時間は 100 試行の平均値を示している. TNSDM は, およそ 500 世代までに解集合がパレートフロントへ向かって急速に収束し, それに要する時間は 10 秒未満であることがわかる. 10,000 世代まで進化させると, パレートフロントの中央部に対して, DP が獲得した解に近い解集合が得られる. ただし, 図中の領域 A については, TNSDM だけでなく, 多種の進化計算によって獲得が困難なパレートフロントの領域として認識されている [46]. TNSDM が 10,000 世代までの最適化に要したのは約 152 秒であり, DP の最適化時間の約 0.6% になる. この結果から, 提案する TNSDM は,  $mk$ -KP 問題に関する事前知識を用いずとも,  $mk$ -KP 問題のパレートフロント中央部に対して, 近似性能の高い解集合を短時間に獲得

表 5.2: 獲得した真の最適解の個数

	アイテム数		
	$n = 100$	$n = 250$	$n = 500$
問題の真の最適解数	122	567	1427
解集団サイズ	200	200	200
獲得した解の数	103	100	105
獲得した真の最適解数	100	5	0

できることがわかる.

## 5.7 真の最適解との比較

本節では離散問題の  $mk$ -KP を用いて最適解と TNSDM で獲得した解とを比較する. 前提として, 進化計算法は確率的探索手法のため, 解く最適化問題における真の最適解を獲得する保証はない. また, 最適解の個数は問題ごとに異なり, 特に連続問題では解が無限に存在する場合がある. これに対して, 進化計算法は, 探索に用いる解集団のサイズ  $N$  を設定し, その数を上限として最適解が作るパレートフロントの近似を試みるものである.

$mk$ -KP において目的数と制約数が等しく ( $m = k$ ), 実行可能率  $\phi = 0.5$  に設定した問題は, 多目的 0/1 ナップザック問題 [26] と呼ばれ, 進化計算の分野でテスト問題として広く用いられており, 真の最適解が求まっているテストデータがいくつか用意されている [47]. テストデータはアイテムの価値と重さ, ナップザックの容量が設定されたものである. このうち, 目的数  $m = 2$ , 制約数  $k = 2$ , アイテム数  $n = \{100, 250, 500\}$  のテストデータを用いて, TNSDM が 1 試行 10,000 世代で獲得する解と真の最適解とを比較する. まず, 表 5.2 にそれぞれの問題における真の最適解と, TNSDM が探索に用いた解集団サイズ  $N$ , 最終世代で獲得した解 (非劣解) の数, さらに, 獲得した解のうち真の最適解であった数を示す. まず, 問題の真の最適解の数は, アイテム数の増加に伴って増加することがわかる. これに対して, TNSDM では解集団サイズ  $N = 200$  を用いて獲得した解の数はいずれも 100 程度であることがわかる. これは TNSDM が解集団中のランク上位半数の解は常に保存しており, 残り半数は常に新たな探索点 (子個体) で置き換わるためである. 次に, 獲得した解が真の最適解であった数を見ると, アイテム数  $n = 100$  の問題では 103 の解のうち 100 が真の最適解であり, パレートフロントの近似だけでなく, 最適解も獲得できたことがわかった. しかし, アイテム数が増加すると獲得した真の最適解の数は大幅に減少し,  $n = 500$  の問題では真の最適解は得られなかった. これは, アイテム数は多くなるほど解空間が大きくなり, 少ない解集団での探索が難しくなるためであ

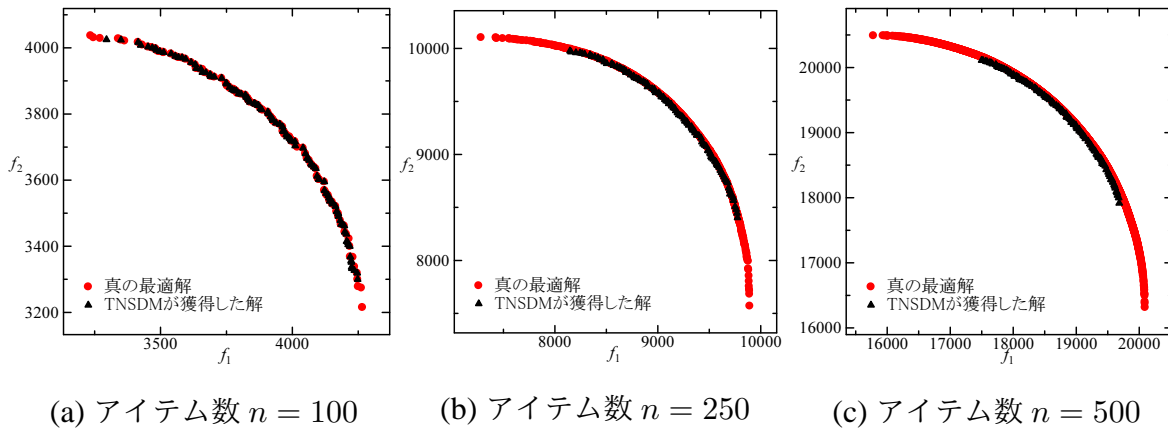


図 5.17:  $mk$ -KP における真の最適解集合と TNSDM が獲得した解集合

る。次に、それぞれの解のパレートフロントの近似性能を見るために、最適解と獲得した解の目的空間におけるプロットを図 5.17 に示す。図 5.17 から、いずれの問題においても TNSDM は真の最適解が作るパレートフロントに対する収束性が高いことがわかる。また、アイテム数の増加に伴って解集団の多様性は低下することがわかる。これらの結果から、TNSDM は  $mk$ -KP において、パレートフロント中央部における近似性能が高い解集合を求めることが可能であることがわかった。また、解空間の小さな問題においてはパレートフロントへの収束性と多様性が高いだけでなく、多数の真の最適解を獲得できることがわかった。

## 5.8 まとめ

制約付き多目的最適化問題に対する MOEA の解探索性能を改善することを目的として、制約違反量と目的関数値による二段階の非支配ソートによって、目的関数値の良好な実行可能解を獲得しやすくする親選択法と、実行不可能解の遺伝子を活用して探索方向への解の収束を促す指向性交配を導入した TNSDM を提案した。連続問題の SRN, TNK, OSY と、離散問題の  $mk$ -KP ( $m = 2$  目的,  $k = \{2, 4, 6, 8, 10\}$  制約, 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$ ) を用いて、TNSDM の解探索性能を従来の CNSGA-II, RTS-O, RTS-C と比較した結果、TNSDM は、従来法と比較してすべての問題において HV の高い解探索性能を示すことが確かめられた。連続問題において、TNSDM は、OSY において POS の多様性が高いことがわかった。また、離散問題の  $mk$ -KP では、得られる POS の収束性と多様性の両面で、従来法より高い解探索性能が実現できることが確かめられた。また、実問題の 3 段式ハイブリッドロケットの概念設計最適化問題において、TNSDM は、従来法の CNSGA-II より短時間で精度の高い POS を獲得できることがわかった。

## 第 6 章

# TNSDM における有用な実行不可能解の選出領域制御

### 6.1 指向性交配の問題 1 : 交配する候補解集合の選出領域

TNSDM における指向性交配が，制約付き多目的最適化問題の進化計算による解法において，解探索性能を高める効果があることが確かめられた．しかし，指向性交配には，解集合  $M$  の選出範囲に関して，2つの問題点がある．

まず，指向性交配は，従来の支配領域によって解集合  $M$  を選出するため，第一の親と探索方向が乖離した解が第二の親になる場合があり，これが解探索の指向性を低下させる問題がある．第一の親との探索方向が近い解を解集合  $M$  に選出することにより，さらに指向性を高めた子生成を実現できる可能性がある．

次に，従来の支配領域では，第一の親を支配する解集合  $M$  が存在しないことがあり，この場合，指向性交配を実行できないため，指向性交配による解探索効果が得られない問題がある．これに対し，解集合  $M$  の選出範囲を拡大することにより，解探索中の指向性交配の数を増加できる可能性がある．

これらの問題を解決するために，本章では，目的空間における第二の親選出領域を，解の支配領域制御法 (Controlling Dominance Areas of Solutions, 以下 CDAS) [22] によって制御する方法を提案する．

### 6.2 TNSDM-CS : 有用な実行不可能解の選出領域を制御する TNSDM

指向性交配による解探索性能の改善効果をさらに高めるために，第二の親候補の選出領域  $M$  を制御する TNSDM-CS (TNSDM with Controlling Selection area) アルゴリズム

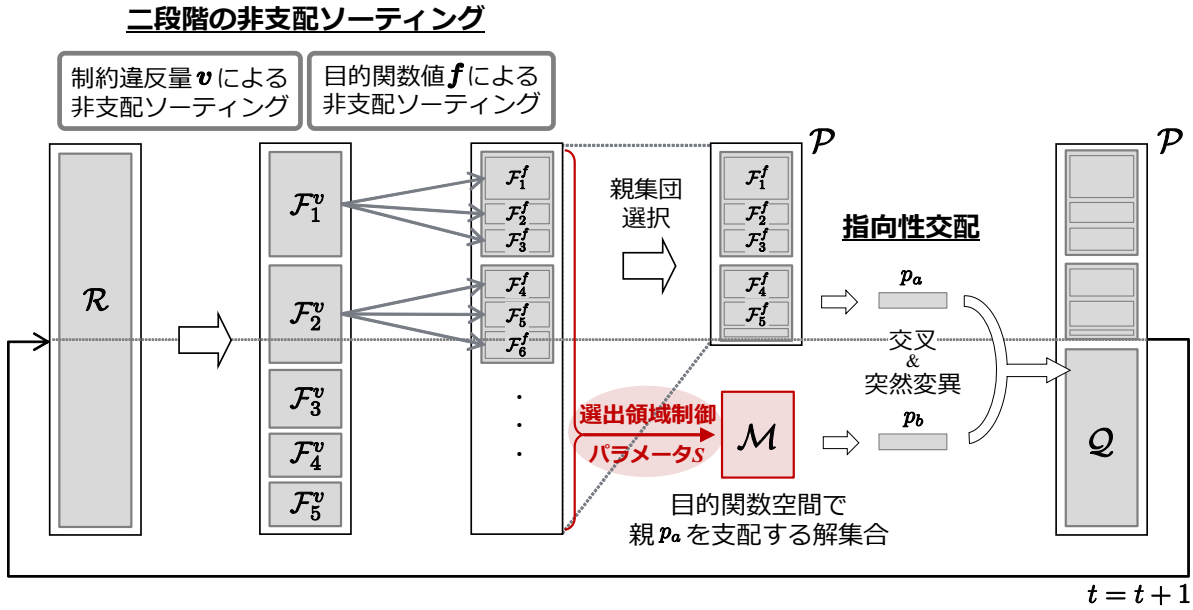


図 6.1: TNSDM-CS のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団)

を提案する。図 6.1 に TNSDM-CS のブロック図を示す。基礎となる TNSDM に対して、ブロック図中の赤色で示した処理を変更する。Algorithm 2 に TNSDM-CS の疑似コードを、用いられる記号を表 6.1 に示す。Algorithm 2 と表 6.1 では、基礎となる TNSDM の記述を黒色で、TNSDM-CS のための記述を赤色で示す。疑似コードの 3–10 行目は二段階非支配ソートによる親集団  $\mathcal{P}$  の選択、12–21 行目は指向性交配と子集団  $\mathcal{Q}$  についての記述である。

### 6.2.1 有用な実行不可能解の選出領域制御法

指向性交配における第二の親候補の選出領域を制御するために、解の支配領域制御法 (CDAS) [22] を導入する。CDAS では解の目的関数値ベクトル  $\mathbf{f}$  を  $\mathbf{f}'$  に変換し、変換した  $\mathbf{f}'$  値に基づいて解同士の支配関係を式 (2.2) を用いて決定する方法である。図 6.2 に CDAS の概念図を示す。解の各目的関数値  $f_i$  は正弦定理を用いた以下の式によって変換される。

$$f'_i(\mathbf{x}) = \frac{r \cdot \sin(\omega_i + S \cdot \pi)}{\sin(S \cdot \pi)} \quad (i = 1, 2, \dots, m), \quad (6.1)$$

ここで、 $r$  は元の目的関数値ベクトル  $\mathbf{f}(\mathbf{x})$  のノルム、 $\omega_i$  は  $f_i$  軸と  $\mathbf{f}(\mathbf{x})$  がつくる角度を示す。また、 $\varphi_i (= S \cdot \pi)$  はユーザ指定のパラメータ  $S$  によって決定される角度であり、これによって解の支配領域を制御することができる。また、その解が支配される領域も同時に制御される。TNSDM-CS では、ユーザパラメータ  $S$  は解探索全体にわたって固定値

---

**Algorithm 2** TNSDM-CS

---

```
1: Randomly generate initial solutions ( $\mathcal{R}$ )
2: for  $t = 1$  to  $T$  do
3:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
4:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sorting } (\mathcal{R}, \succ_v)$ 
5:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
6:      $j = 1 + |\mathcal{F}^f|$ 
7:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sorting } (\mathcal{F}_i^v, \succ_f)$ 
8:   end for
9:   Crowding distance ( $\mathcal{F}^f$ )
10:   $\mathcal{P} = \text{Truncation } (\mathcal{R}, |\mathcal{R}|/2)$ 
11:  Calculate  $f'$  from  $f$  ( $\mathcal{R}, S$ )
12:  for  $j = 1$  to  $|\mathcal{Q}|$  do
13:     $p_a = \text{Tournament selection } (\mathcal{P})$ 
14:     $\mathcal{M} = \{\mathbf{x} \in \mathcal{R} \mid \mathbf{x} \succ_{f'} p_a\}$ 
15:    if  $p_a$  is feasible and  $|\mathcal{M}| \geq 2$  then
16:       $p_b = \text{Tournament selection } (\mathcal{M}) : \textit{Directed Mating}$ 
17:    else
18:       $p_b = \text{Tournament selection } (\mathcal{P}) : \textit{Conventional Mating}$ 
19:    end if
20:     $Q_j = \text{Crossover and Mutation } (p_a, p_b)$ 
21:  end for
22:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
23: end for
24:  $\text{POS} = \{\mathbf{x} \in \mathcal{R}^{\text{feasible}} \mid \neg \exists \mathbf{y} \in \mathcal{R}^{\text{feasible}} : \mathbf{y} \succ_f \mathbf{x}\}$ 
```

---

を用いることとする。ここで、 $f'_i(\mathbf{x})/\sin(\pi - \varphi_i - \omega_i)$  は  $r/\sin(\varphi_i)$  と等しいため、変換された目的関数値  $f'_i(\mathbf{x})$  はパラメータ  $S$  によって指定される角度  $\varphi_i (= S \cdot \pi)$  を用いて計算が可能である。

図 6.3, 6.4, 6.5 は、それぞれ異なるパラメータ  $S$  を用いた場合の解  $\mathbf{x} (= p_a)$  の CDAS による目的関数値の変換と解  $\mathbf{x}$  の支配領域と、被支配領域すなわち第二の親集団  $\mathcal{M}$  の選出領域を示している。図 6.3 は  $S = 0.5$  ( $\varphi_i = \pi/2$ ) の場合を示している。このとき、解の目的関数値は変化せず ( $f'_i(\mathbf{x}) = f_i(\mathbf{x})$ )、従来の支配領域と  $\mathcal{M}$  の選出領域に等しい。図 6.4 に示されるように  $S < 0.5$  ( $\varphi_i < \pi/2$ ) とすると、目的関数値はもとより大きな値になり ( $f'_i(\mathbf{x}) > f_i(\mathbf{x})$ )、目的空間における解の支配領域と  $\mathcal{M}$  の選出領域が拡大する。一方、

表 6.1: 擬似コード内で使用される記号 (TNSDM-CS)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$S$	第二の親の候補解集合 $\mathcal{M}$ の選出領域を制御するパラメータ
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

図 6.5 に示されるように  $S > 0.5$  ( $\varphi_i > \pi/2$ ) とすると, 目的関数値はもとより小さな値になり ( $f'_i(\mathbf{x}) < f_i(\mathbf{x})$ ), 解の支配領域と  $\mathcal{M}$  の選出領域が縮小する. TNSDM-CS では, 指向性交配において, この制御された  $\mathcal{M}$  の選出領域から第二の親を選択する. 以降, 変更した目的関数値  $f'$  において, 解  $\mathbf{x}$  が  $\mathbf{y}$  を支配するときの表記を,  $\mathbf{x} \succ_{f'} \mathbf{y}$  と定義する.

TNSDM-CS では, 親集団  $\mathcal{P}$  選択の後, 全解集団  $\mathcal{R}$  の解の目的関数値を式 (6.1) を用いて変換する (Algorithm 2 11 行目). ここで, 留意すべきは CDAS による目的関数値の変更は, 指向性交配における親ペア選択のみに用いることであり, 二段階の非支配ソーティングにおいては元の目的関数値  $f$  による従来の支配 ( $\succ_f$ ) を用いて解をフロント分類する. 指向性交配において, 変換された目的関数値  $f'$  を用いた支配 ( $\succ_{f'}$ ) に基づいて第二の親候補集団  $\mathcal{M}$  を選出する.

従来, CDAS による目的関数値の変換は, 支配ベースの MOEA を用いて制約条件のない 4 目的以上の多数目的最適化問題を解くために提案された. 親集団選択における解の非支配ソーティングの為に用いられ, 解の精緻なランキングを実現する. 一方, 本研究では, 制約付き多目的最適化問題の解法のため, 指向性交配において親のペア選択に用いる点で異なる. 拡大した支配領域では, 解の探索方向への指向性は低下してしまうが, 第一の親を支配可能な領域が拡大するため, 第二の親候補が獲得しやすくなる. また, 縮小した支配領域では, 第一の親の探索方向に近い解のみが第二の親候補となるため, 解探索の



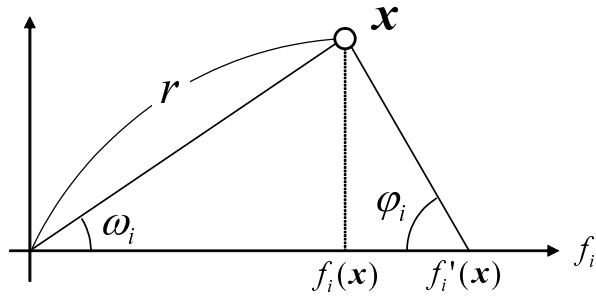


図 6.2: CDAS による目的関数値の変換

指向性が高まることが期待できる。

## 6.2.2 期待される効果

図 6.3 は  $S = 0.5$  における第一の親  $p_a$  の第二の親選出領域を示している。この場合、第二の親候補集団  $M$  の選出領域は従来のものに等しく、従来の指向性交配と同じである。図 6.3 では、4 つの解が  $M$  として選択されている。また、図 6.4 は  $S < 0.5$  の場合をあらわしている。この場合、 $M$  の選出領域が従来の領域よりも拡大される。それによって、 $p_a$  の解探索方向の指向性が弱まるが、選出領域に解が存在する可能性が高まる。解集団中に高い目的関数値を持つ実行不可能解が少ない場合などにおいて、より多くの第一の親が交配相手を獲得しやすくなるため、指向性交配を実行できる回数が増え、解探索性能が高まることが期待できる。一方で、図 6.5 は  $S > 0.5$  の場合を表している。この場合、 $M$  の選出領域が従来の領域よりも縮小される。そのため、 $p_a$  の解探索方向の指向性が強まり、解探索性能向上が期待できる。しかし、縮小した  $M$  の選出領域には解が含まれにくくなるため、いくつかの第一の親が指向性交配を実行できなくなる可能性も高まる。

## 6.3 実験内容

有用な実行不可能解の選出領域制御法を TNSDM に導入した TNSDM-CS において、パラメータ  $S$  を変化させた場合の指向性交配の実行回数、解探索性能を  $mk$ -KP と mCDTLZ 問題を用いて検証する。また、解探索性能は TNSDM と CNSGA-II と比較する。

### 6.3.1 パラメータ

$mk$ -KP の問題パラメータは、アイテム数  $n = 500$ 、目的数  $m = \{2, 4, 6\}$ 、制約数  $k = 6$ 、実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  に設定した。 $mk$ -KP におけるアルゴリズムの遺伝的パラメータは、一様交叉法 (交叉率  $P_c = 1.0$ )、ビット反転突然変異法 (変異率

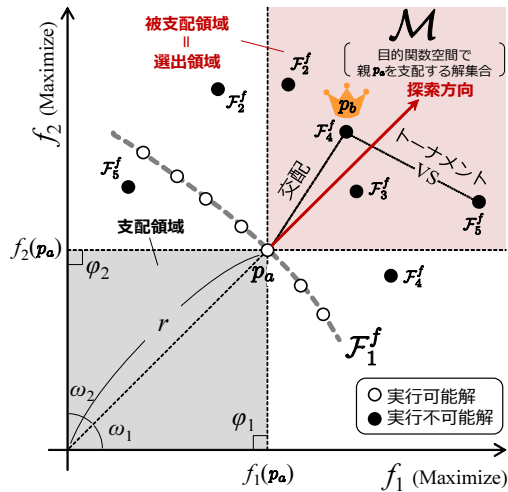


図 6.3: 従来の選出領域 ( $S = 0.5$ )

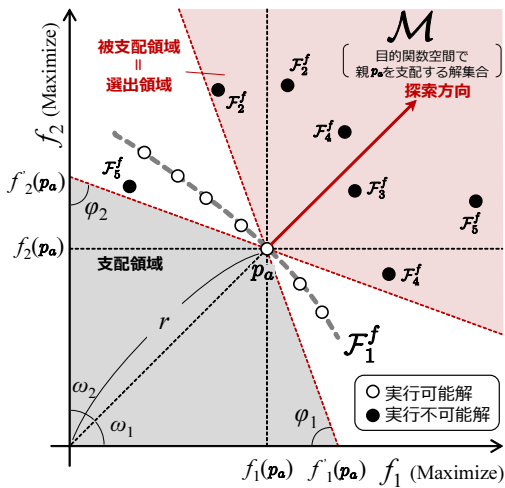


図 6.4: 拡大した選出領域 ( $S < 0.5$ )

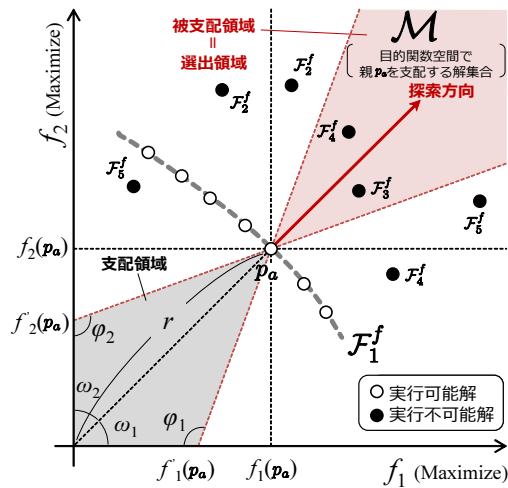


図 6.5: 縮小した選出領域 ( $S > 0.5$ )

$P_m = 1/n$ ), 世代数  $T = 10,000$  を用いた. また, mCDTLZ の問題パラメータは, 変数の数  $n = 12$ , 目的数 (=制約数)  $m = \{2, 4, 6\}$  に設定した. この問題における遺伝的パラメータは, SBX 交叉法 (交叉率  $P_c = 0.8$ ,  $\eta_c = 15$ ), Polynomial mutation 突然変異法 (変異率  $P_m = 0.2$ ,  $\eta_m = 20$ ), 世代数  $T = 5,000$  を用いた.  $mk$ -KP と mCDTLZ のいずれにおいても集団サイズ  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) を用いた. 実験の結果は 100 試行の平均値を用いて示す.

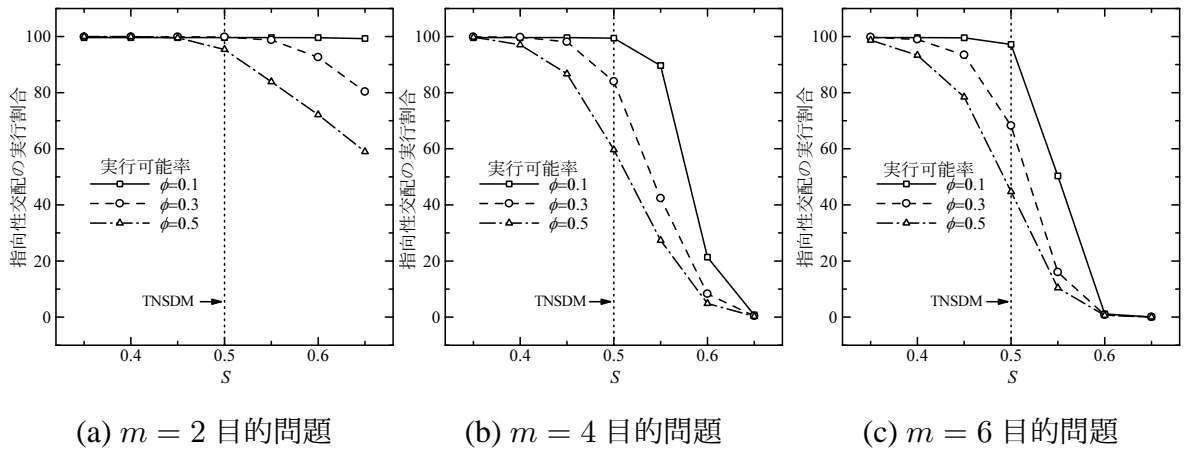


図 6.6:  $mk$ -KP における指向性交配の実行割合

### 6.3.2 評価尺度

獲得した解集合を総合評価する Hypervolume ( $HV$ ) によって TNSDM-CS の効果を検証する.  $HV$  の算出において, 参照点はそれぞれ  $mk$ -KP では  $\mathbf{r} = (0, 0, \dots, 0)$ , mCDTLZ では  $\mathbf{r} = (1.1, 1.1, \dots, 1.1)$  を用いた. その後, 多様性と収束性を, それぞれ Maximum Spread ( $MS$ ), Norm を用いて詳細に評価する.

## 6.4 実験結果と考察

### 6.4.1 $mk$ -KP における $\mathcal{M}$ の選出領域制御と指向性交配の回数

TNSDM-CS の解探索性能を検証する前に, まず  $\mathcal{M}$  の選出領域をパラメータ  $S$  によって変化させたときの指向性交配の実行回数について調べる. TNSDM および TNSDM-CS では, 子を生成するときに指向性交配だけでなく従来の交配を行う場合がある. 選択した第一の親が実行可能解かつ交配する候補解が 2 つ以上得られた場合 ( $|\mathcal{M}| \geq 2$ ), 第二の親を  $\mathcal{M}$  から選択する指向性交配を行うが, そうでない場合は親集団  $\mathcal{P}$  から選択する従来の交配を行う. 図 6.6 は,  $mk$ -KP において解探索中のすべての交配のうち指向性交配を実行した回数の割合を百分率で示している. 図中に垂直の点線で示される  $S = 0.5$  の結果は, 選出領域制御を用いない TNSDM の結果に等しく,  $S < 0.5$  は従来よりも拡大した選出領域,  $S > 0.5$  は縮小した選出領域の結果を示している.

これらの結果から, まず, 指向性交配の回数は,  $S$  の値が大きくなるにつれて減少することがわかる. これは,  $\mathcal{M}$  の選出領域が縮小されるほど解が  $\mathcal{M}$  に含まれにくくなり, 指向性交配が実行できない回数が増えることを示している. 一方で,  $S$  の値を 0.5 以下とし

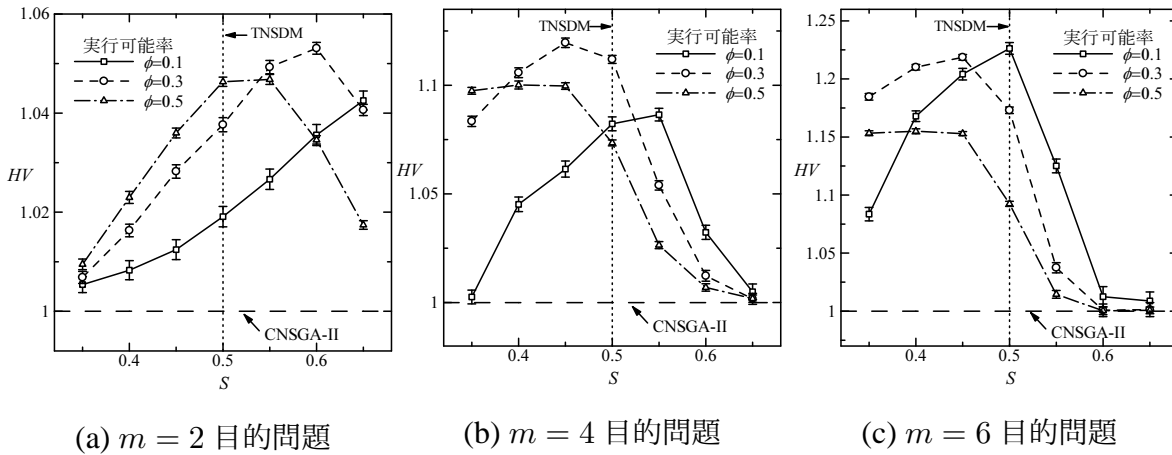


図 6.7:  $mk$ -KP における最終世代の  $HV$

て選出領域を拡大すると、従来よりも指向性交配の回数が増加することがわかる。これらの結果から、TNSDM-CS においてパラメータ  $S$  を変化させることによって指向性交配の回数を変化可能であることが明らかになった。

また、 $mk$ -KP の実行可能率  $\phi$  と目的関数の数も指向性交配の回数に影響を与えることがわかる。実行可能率が減少すると指向性交配の回数が増加する。これは、実行可能率の低い制約の厳しい問題では、生成される子個体が実行不可能解になりやすいため、解集団中に親集団  $\mathcal{P}$  を支配可能な実行不可能解の数が増えるためだと考えられる。また、目的数が増えると、指向性交配の回数が増えるほど、支配の条件を満たすことが難しくなるためと考えられる。

#### 6.4.2 $mk$ -KP における $\mathcal{M}$ の選出領域制御による解探索性能

指向性交配における解集合  $\mathcal{M}$  の選出領域をパラメータ  $S$  で変化させた場合の最終世代の  $HV$  を図 6.7 に示す。また、エラーバーは 95% 信頼区間を示している。従来法 CNSGA-II で得られた結果も水平線で示し、すべての結果は CNSGA-II によって得られた  $HV$  を 1.0 として正規化している。また、 $S = 0.5$  の結果は、TNSDM の結果に等しい。

これらの結果から、パラメータ  $S$  を変化させて  $\mathcal{M}$  の選出領域を適切に変化させることによって、TNSDM よりも高い  $HV$  値を示す解集合を獲得できることがわかった。また、本実験で用いたすべての  $mk$ -KP において、TNSDM-CS はいずれの  $S$  を用いても CNSGA-II よりも高い  $HV$  を示した。以下結果について詳細に述べる。

まず、図 6.7 (a) に示される 2 目的の  $mk$ -KP における結果について議論する。実行可能率  $\phi = 0.1$  では、 $S$  の増加に伴って  $HV$  が単調に向上することがわかる。このとき、図 6.6 (a) の実行可能率  $\phi = 0.1$  を見ると、 $S$  によらずほぼ 100% の子個体が指向性交配に

よって生成されていることがわかる。これらの結果から、実行可能率  $\phi$  が低く、解集団中に実行不可能解が多い場合は、 $S$  を大きくして解集合  $\mathcal{M}$  の選出領域を縮小し、指向性を高めた交配法が  $HV$  の高い解探索性能を達成するといえる。次に、図 6.7 (a) の実行可能率  $\phi = 0.3$  の結果を見ると、 $HV$  を最大化する最適値  $S^*$  が存在し、 $S^* = 0.6$  であることがわかる。つまり、 $S$  は大きいほど良いわけではなく、 $S > 0.6$  では  $HV$  が低下してしまう。これについて、図 6.6 (a) の実行可能率  $\phi = 0.3$  の結果を見ると、 $S \geq 0.5$  では選出領域が縮小するため、指向性交配の実行回数が減少している。これが  $HV$  の低下の原因になると考えられる。すなわち、 $S$  の増加は交配の指向性を高める効果がある一方で、指向性交配の実行回数を減少させてしまう。そのため、両者のバランスが良い最適な  $S^*$  を用いる場合に  $HV$  が最大化するといえる。

次に、4, 6 目的問題について、図 6.7 (b) と (c) の実行可能率が高い  $\phi = \{0.3, 0.5\}$  の場合、 $S^* < 0.5$  となり、解集合  $\mathcal{M}$  の選出領域を拡大した場合に  $HV$  が高まることわかる。図 6.6 に示すように、目的数  $m$  が多く、実行可能率  $\phi$  が大きいほど、指向性交配を実行できる回数は減少する。そのため、小さな  $S$  によって解集合  $\mathcal{M}$  の選出領域を拡大し、指向性交配の実行回数を増やすことによって、 $HV$  が改善したと考えられる。

最後に、すべての目的数  $m$  の問題において、実行可能率  $\phi$  の増加に伴って最適な  $S^*$  の値が小さくなることがわかる。実行可能率  $\phi$  が低い問題では、解集団中の実行不可能解の数が多いため、指向性交配を実行できる回数が多くなる。このときは、 $S$  を大きくして交配の指向性を高めた方が  $HV$  が高まるため、 $S^*$  は比較的高くなる。逆に実行可能率  $\phi$  が高い問題では、解集団中の実行不可能解の数が少ないため、 $S$  を大きくし過ぎると指向性交配の実行回数が減少して  $HV$  が低下するため、 $S^*$  は比較的低い値になると考えられる。

このように、図 6.7 に示す全ての結果において、最小の  $S$  によって選出領域を最大限に拡大し、実行不可能解の選択肢を増やした場合より、 $S$  を増加して選出領域を縮小し、第一の親より全ての目的関数値がバランス良く良好な限られた実行不可能解のみを選択できる場合に、 $HV$  が向上することがわかった。実行不可能解を親にする他の方法として、制約条件を緩和して解く方法がある。第一の親をある実行可能解とした場合、緩和法は、実行不可能解を第二の親として選択する可能性があるが、その目的空間における位置は考慮しない。全く同じとはいえないが、これは、指向性交配において、最小の  $S$  によって選出領域を最大化した状態で第二の親を選択することに近い。指向性交配では、最小の  $S$  より、大きな  $S$  によって解探索性能が改善することから、制約緩和法のように単純に実行不可能解を親にするだけでなく、指向性交配のように実行可能解ごとに交配する適切な実行不可能解を取捨選択することが解探索を促進するといえる。

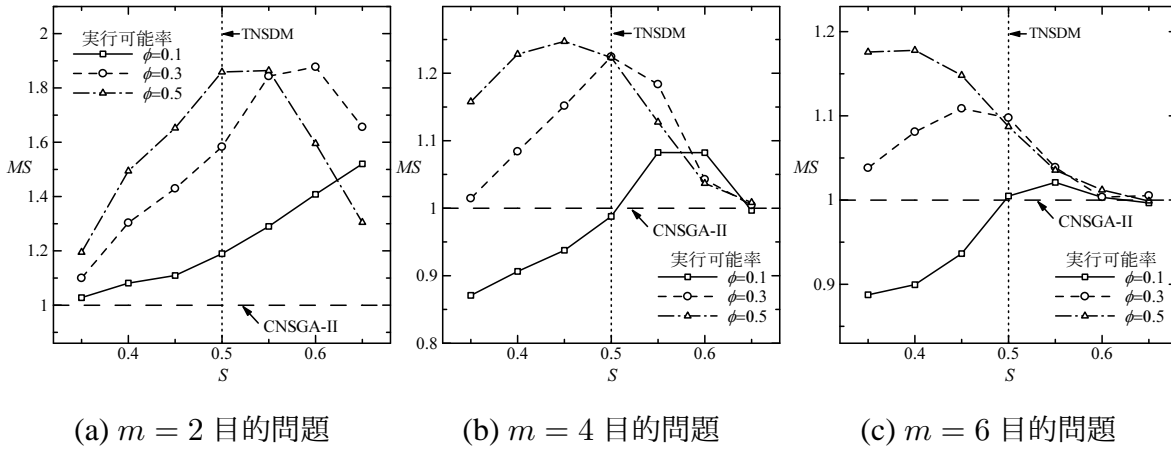


図 6.8:  $mk$ -KP における最終世代の  $MS$

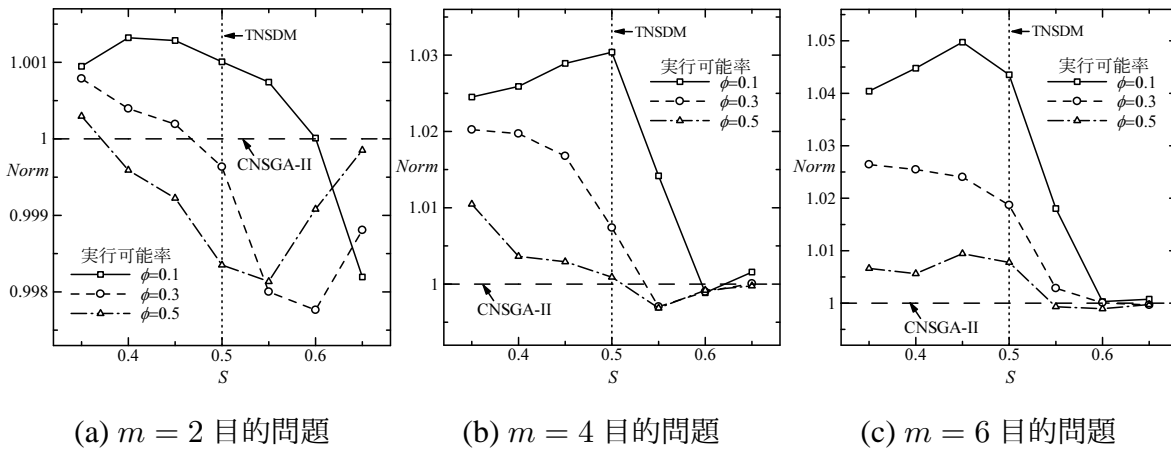


図 6.9:  $mk$ -KP における最終世代の  $Norm$

### 6.4.3 $mk$ -KP における $\mathcal{M}$ の選出領域の変化に対する解の多様性と収束性

$HV$  の改善に貢献した要因を解析するために、解集合の目的空間における多様性と、真の最適解集合への収束性を独立して評価する。得られた解集合の多様性について  $MS$  の結果を前節と同様の形式で図 6.8 に示す。これらの結果から、 $S$  を適切に変化させることによって、 $MS$  値はすべての問題において顕著に改善されることがわかる。また、 $MS$  を最大化する  $S$  は、 $HV$  を最大化する  $S^*$  と等しいか、より大きい値であることがわかる。次に、得られた解集合の収束性について  $Norm$  の結果を図 6.9 に示す。 $Norm$  を最大化する  $S$  は、 $HV$  を最大化する  $S^*$  より小さい値になる傾向がある。

この結果から、良い  $MS$  と  $Norm$  をバランスさせる  $S^*$  によって最も高い  $HV$  が示さ

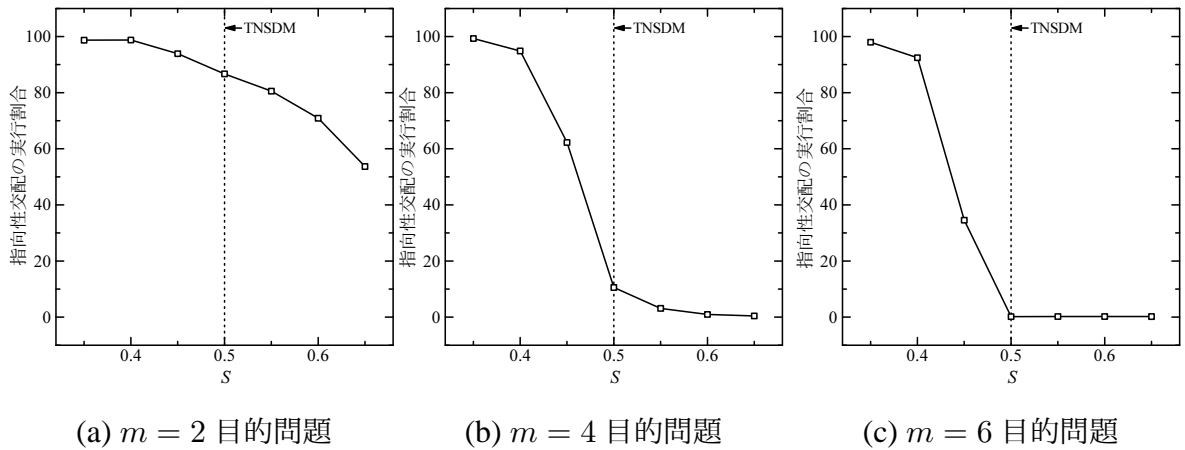


図 6.10: mCDTLZ 問題における指向性交配の実行割合

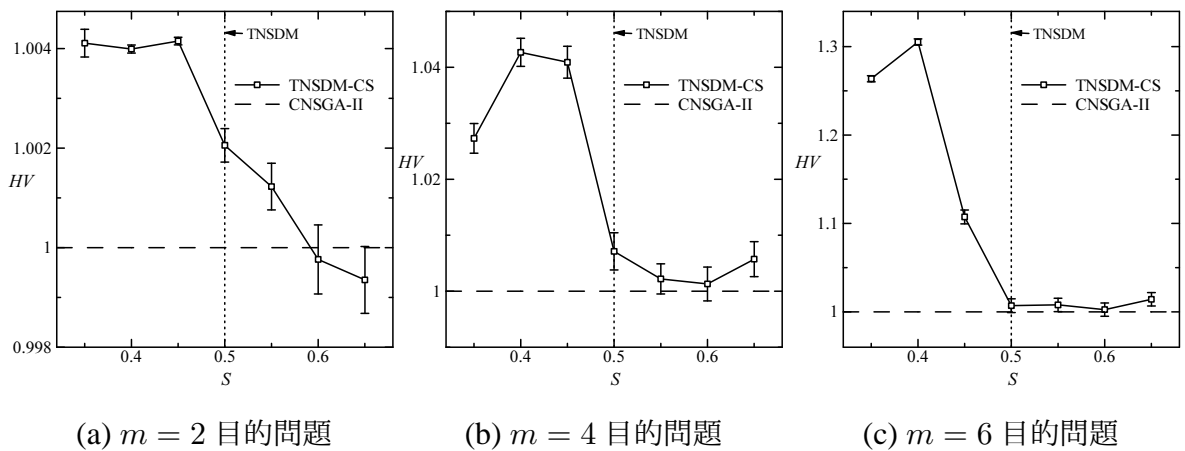


図 6.11: mCDTLZ 問題における最終世代の  $HV$

れたと考えられる。また、 $S$  を変化させた場合の  $HV$  と  $MS$  の変化の傾向が似ていることから、 $mk$ -KP において  $HV$  の改善に  $MS$  の改善が大きく貢献していると考えられる。

#### 6.4.4 mCDTLZ における $\mathcal{M}$ の選出領域制御と指向性交配の回数

mCDTLZ 問題における結果について、まず、 $\mathcal{M}$  の選出領域をパラメータ  $S$  によって変化させたときの指向性交配の実行回数について調べる。6.4.1 節と同様に、図 6.10 に mCDTLZ 問題において解探索中のすべての交配において指向性交配を実行した回数の割合を百分率で示している。

これらの結果から、特に  $m = \{4, 6\}$  目的問題において TNSDM は指向性交配をほとんど実行できていないことがわかる。しかし、パラメータ  $S$  を小さくし、 $\mathcal{M}$  の選択領域を拡大することによって指向性交配の実行割合を 100% 付近まで増加できている。

#### 6.4.5 mCDTLZ における $\mathcal{M}$ の選出領域制御による解探索性能

指向性交配における解集合  $\mathcal{M}$  の選出領域をパラメータ  $S$  で変化させた場合の最終世代の  $HV$  を図 6.11 に示す。従来法の CNSGA-II で得られた結果も水平線で示し、すべての結果は CNSGA-II によって得られた  $HV$  を 1.0 として正規化している。また、エラーバーは 95% 信頼区間を示している。これらの結果から、まず、TNSDM( $S = 0.5$ ) は従来法の CNSGA-II よりもすべての問題ケースにおいて高い  $HV$  を示すことがわかる。mCDTLZ は、TNK や OSY のように変数空間における POS が実行可能領域と不可能領域の境界に存在する問題である。この実験結果からも、TNSDM が、実行可能領域と不可能領域の境界に存在する POS の獲得において、従来の CNSGA-II より高い探索性能を示すことが確かめられた。また、mCDTLZ は、凹型のパレートフロントを有する問題であることから、TNSDM は、凹型のパレートフロントに対しても頑健な解探索性能を示すことが確かめられた。次に、パラメータ  $S$  を 0.5 よりも小さくし、拡大した  $\mathcal{M}$  の選択領域を用いることでさらに  $HV$  が高まることがわかる。また、 $mk$ -KP の結果と同様に  $HV$  を最大化する最適なパラメータ  $S^*$  が存在し、 $m = \{2, 4, 6\}$  目的問題に対してそれぞれ  $S^* = \{0.45, 0.4, 0.4\}$  であった。また、目的数が大きい問題ほど従来指向性交配が行えていなかった分、 $HV$  値の改善が顕著になっている。したがって、mCDTLZ 問題において、 $S < 0.5$  に設定した TNSDM において指向性交配の回数を増加させることで解探索性能が向上することがわかった。

#### 6.4.6 解探索性能と指向性交配の回数との関係

TNSDM-CS においてパラメータ  $S = \{0.35, 0.40, \dots, 0.65\}$  としたときの解探索性能 ( $HV$ ) と指向性交配の実行割合の関係について分析する。図 6.12 と図 6.13 にそれぞれ  $mk$ -KP と mCDTLZ 問題における結果を示す。なお、 $HV$  の結果は CNSGA-II より得られた結果を 1.0 として正規化したものを用いている。

これらの結果から、実験に用いたすべての問題において、指向性交配の実行割合がおおよそ 90% のあたりで  $HV$  値が最も高くなることがわかる。すなわち、パラメータ  $S$  を指向性交配の実行割合を 90% を満たすように制御することで、指向性交配の効果が最も高まることを意味している。

#### 6.4.7 TNSDM-CS の効果と問題

$mk$ -KP と mCDTLZ 問題を用いた TNSDM-CS の性能検証の結果、適切なパラメータ  $S^*$  を設定した TNSDM-CS が、TNSDM と等しいかそれ以上の解探索性能を示すことが



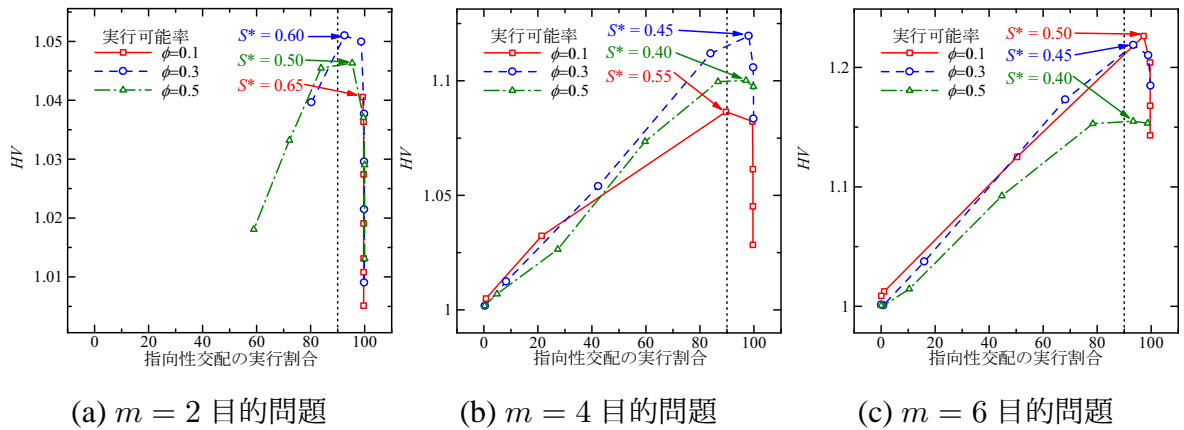


図 6.12:  $mk$ -KP における解探索性能 ( $HV$ ) と指向性交配の実行回数の割合

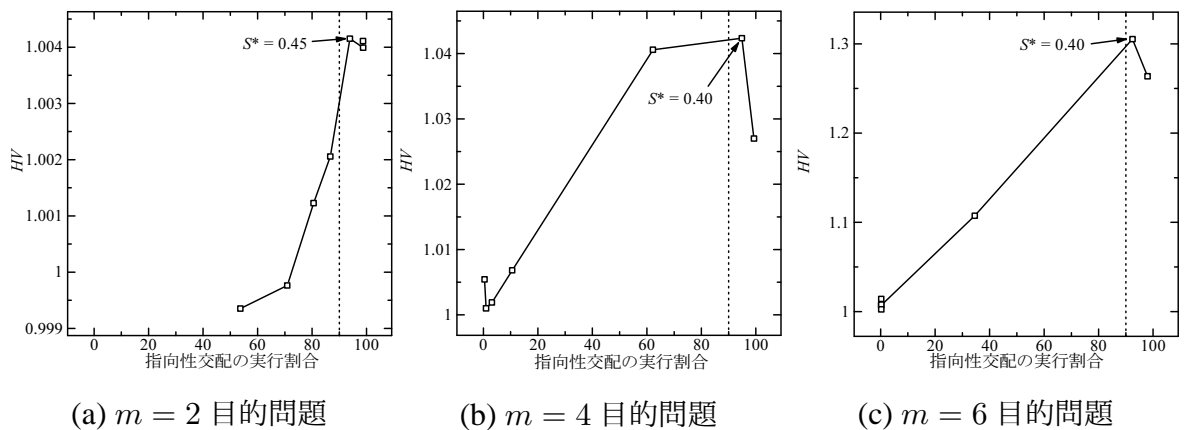


図 6.13:  $mCDTLZ$  問題における解探索性能 ( $HV$ ) と指向性交配の実行回数の割合

わかった。しかし、最も解探索性能が高まるパラメータ  $S^*$  は、 $CMOP$  ごとに異なるため、 $S$  を少しずつ変化させながら  $TNSDM-CS$  を繰り返し実行して実験的に見出す他ない。指向性交配において、解の選出領域制御法は解探索性能を高める効果があるが、その適切なパラメータ  $S^*$  は時間をかけてチューニングせざるを得ないところに問題がある。

## 6.5 解の選出領域の適応制御

### 6.5.1 コンセプト

指向性交配における選出領域制御法のパラメータ  $S$  のチューニングを回避するために、一度の解探索中に  $S$  を適応的に決定する  $TNSDM-ACS$  ( $TNSDM$  with Adaptive Controlling Selection areas) を提案する。

解の選出領域制御法では、 $S$  を大きくして  $\mathcal{M}$  の選出領域を縮小するほど解探索の指向性が高まる一方で、 $\mathcal{M}$  に解が得られにくくなり指向性交配の実行回数が減少する。反対

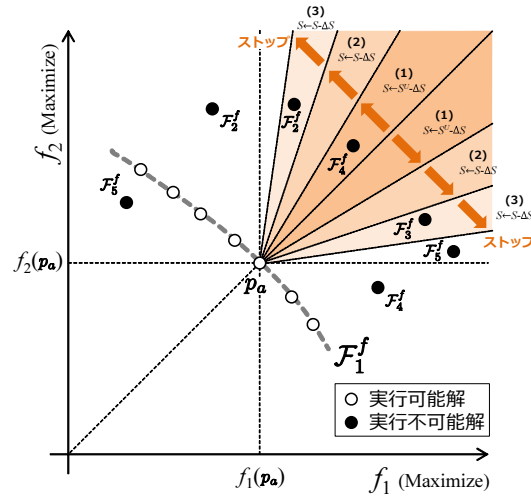


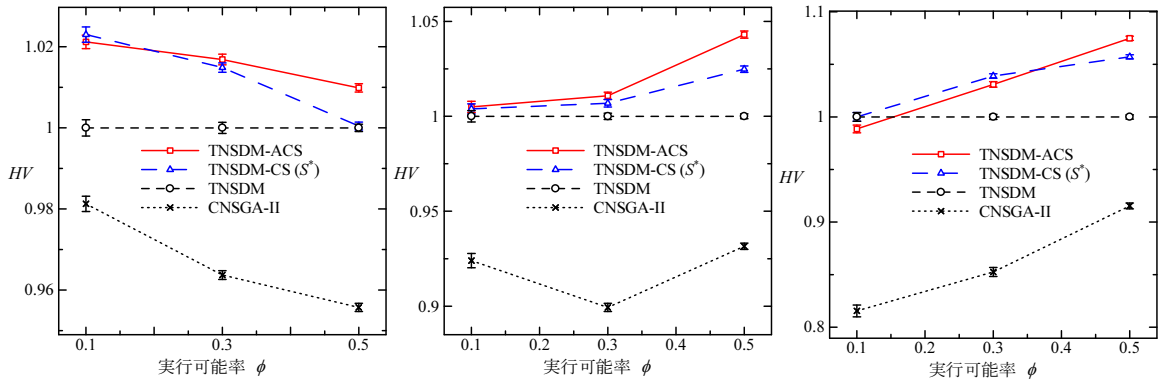
図 6.14: 解の選出領域の適応制御の概念図

に， $S$  を小さくし選出領域を拡大するほど  $\mathcal{M}$  に解が得られやすくなり指向性交配の実行回数が増加する一方で，解探索の指向性が弱まる．6.4.6 節の結果から，約 90% 以上の親  $p_a$  が指向性交配を実行できる範囲で，最も大きな  $S$  を設定した場合に，高い解探索性能を達成できることが明らかになっている．そのため，TNSDM-ACS は，各親  $p_a$  が  $\mathcal{M}$  に解が 2 つ以上含まれる範囲で最大の  $S$  を適応的に設定することとする．TNSDM-CS の概念図を図 6.14 に示す．

## 6.5.2 方法

上記の設計思想を実現する方法としては，様々やり方が考えられるが，今回は以下の単純な手順で  $S$  を適応的に設定する．ひとつの子を生成するために，まず，親集団  $\mathcal{P}$  から第一の親  $p_a$  をバイナリトーナメント選択によって選択する．次に，第二の親  $p_b$  の候補解集合  $\mathcal{M}$  を選出するために， $S$  を最大値  $S^U$  に設定し ( $S \leftarrow S^U$ )，最も縮小した選出領域を用いて  $\mathcal{M}$  の選出を試みる．その結果， $|\mathcal{M}| < 2$  の場合は  $|\mathcal{M}| \geq 2$  を満たすまで，もしくは， $S$  が最小値  $S^L$  になるまで， $S$  から  $\Delta S$  を減じ ( $S \leftarrow S - \Delta S$ )，選出領域を徐々に拡大しながら  $\mathcal{M}$  の選出を繰り返す．図 6.14 では，3 回  $\Delta S$  を減じたときに， $|\mathcal{M}| \geq 2$  を満たし， $\mathcal{M}$  の選出を終える例を示している．

その後は，従来の指向性交配と同様に， $|\mathcal{M}| \geq 2$  の場合， $\mathcal{M}$  から混雑度トーナメント選択によって第二の親  $p_b$  を選択する．また，親  $p_a$  が実行不可能解の場合と， $\mathcal{M}$  のサイズが 1 以下 ( $|\mathcal{M}| \leq 1$ ) の場合，指向性交配は実行できず，従来法の CNSGA-II などと同様に，親集団  $\mathcal{P}$  から混雑度トーナメント選択によって第二の親  $p_b$  を選択する．この操作をすべての親  $p_a$  に対して行う．



(a) 目的数  $m = 2$                       (b) 目的数  $m = 4$                       (c) 目的数  $m = 6$

図 6.15: 選出領域の適応制御法による最終世代の  $HV$  (総合評価)

TNSDM-CS では、解探索開始時に指定された  $S$  をすべての親  $p_a$  に対して用いるのに対し、TNSDM-ACS では、親  $p_a$  ごとに異なる  $S$  が使用される。

### 6.5.3 適応制御の効果

TNSDM-ACS における解の選出領域の適応制御法の効果を検証するために、TNSDM、TNSDM-CS の解探索性能を  $mk$ -KP 問題を用いて比較する。また、従来法の CNSGA-II の結果も合わせて掲載する。TNSDM-CS では、 $S = \{0.25, 0.3, \dots, 0.75\}$  を用いた事前実験の結果、最も高い  $HV$  を達成した最適な  $S^*$  を用いた場合の結果を示す。各問題における TNSDM-CS の最適な  $S^*$  を表 6.2 に示す。TNSDM-ACS では、 $S$  の最大値  $S^U = 0.70$ 、最小値  $S^L = 0.25$  とし、 $\Delta S = 0.05$  の間隔で  $S$  を変更する。

目的数  $m$  と実行可能率  $\phi$  を変化させた  $mk$ -KP において、最終世代に得られた  $HV$  の結果を図 6.15 に示す。ここで、図 6.15 は、TNSDM の結果を 1.0 として正規化してプロットしている。また、エラーバーは 95% 信頼区間を示している。

まず、すべての問題において、TNSDM は CNSGA-II より高い  $HV$  を示すことがわかる。次に、固定値の  $S^*$  を用いる TNSDM-CS は、TNSDM と等しいか、それ以上の  $HV$  を示すことがわかる。 $m = 6$  目的、実行可能率  $\phi = 0.5$  の問題では、 $S^* = 0.5$  であるため、TNSDM と TNSDM-CS によって得られた結果が等しい。さらに、パラメータチューニングを必要としない TNSDM と TNSDM-ACS を比較すると、 $m = 6$  目的、実行可能率  $\phi = 0.1$  の問題を除くすべての問題で、TNSDM-ACS が TNSDM より高い  $HV$  を示すことがわかった。

最後に、固定値の  $S^*$  を用いる TNSDM-CS と TNSDM-ACS を比較する。 $S^*$  を用いる TNSDM-CS の結果は、事前実験でパラメータチューニングした最良値であるため、本来、これらを公平に比較できないことに注意されたい。 $m = 2$  目的、実行可能率  $\phi = 0.1$  と

表 6.2: TNSDM-CS における最適な  $S^*$

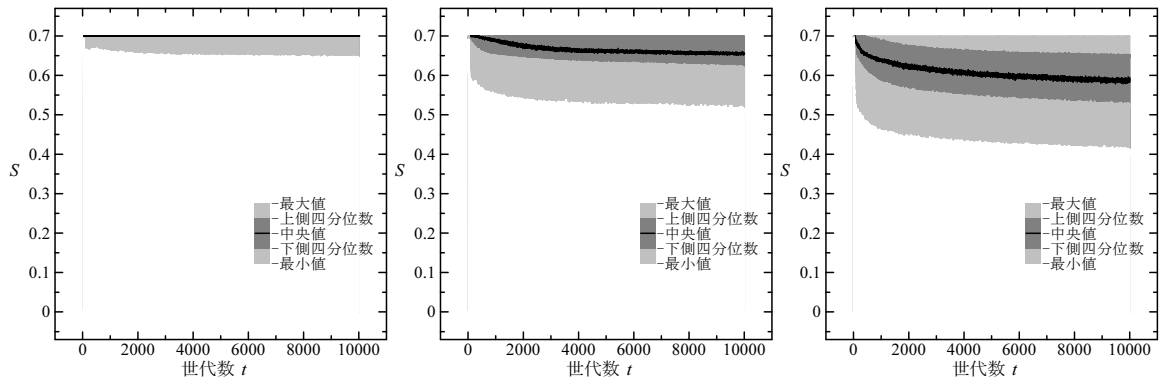
		目的数 $m$		
		2	4	6
実行可能率 $\phi$	0.1	0.65	0.55	0.50
	0.3	0.60	0.45	0.45
	0.5	0.55	0.40	0.40

$m = 6$  目的, 実行可能率  $\phi = \{0.1, 0.3\}$  の問題において, TNSDM-ACS は TNSDM-CS より  $HV$  が低い, その他の問題では, TNSDM-ACS が TNSDM-CS より高い  $HV$  を示すことがわかる. 特に実行可能率  $\phi$  が高い問題において, TNSDM-ACS は, 時間をかけてチューニングした  $S^*$  を用いる TNSDM-CS より高い解探索性能を示すことがわかった. これは, TNSDM-ACS が, 各親  $p_a$  ごとにパラメータ  $S$  を決定することによる効果であると考えられる. TNSDM-CS では, すべての親  $p_a$  が同じ  $S$  を用いて解集合  $M$  を選出しようとする. そのため, 解の目的空間における分布によっては, 解集合  $M$  が得られない親  $p_a$  や,  $M$  に多数の解が存在し, さらに選出領域を縮小できる親  $p_a$  も存在する. TNSDM-ACS は, それぞれの親  $p_a$  に応じて選出領域を適応的に決定できるため, TNSDM-CS より解探索性能が改善したと考えられる.

TNSDM-ACS の適応制御によって決定されたパラメータ  $S$  について,  $m = 2$  目的, 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の問題を用いて解析する. 各世代の指向性交配において用いられた  $S$  の五数要約の 100 試行の平均値を図 6.16 に示す.

まず, 図 6.16 (a) の結果から, 実行可能率  $\phi = 0.1$  の問題では, 多くの指向性交配において  $S = 0.7$  が用いられていることがわかる. そのため, 図 6.7 において, 固定値の  $S^*$  を用いる TNSDM-CS と TNSDM-ACS が, ほぼ同等の  $HV$  を示したと考えられる.

次に, 図 6.16 (b) と (c) の結果から, 問題の実行可能率  $\phi$  の増加に伴って, 交配ごとに用いられる  $S$  が多様化することがわかる. 図 6.16 (c) の実行可能率  $\phi = 0.5$  の問題において, TNSDM-CS では, 最適なパラメータは  $S^* = 0.55$  であるのに対して, TNSDM-ACS では, 少なくとも半数の親  $p_a$  が,  $S = 0.55$  より縮小した選出領域を用いて指向性交配を実行していることがわかる. また,  $S = 0.55$  で交配相手を獲得できない場合は, より拡大した選出領域を用いることで, 指向性交配を実行していることがわかる. このように, TNSDM-ACS は, 交配ごとの状況に合わせて  $S$  を変更することによって, 固定値の  $S$  を用いる TNSDM-CS より高い  $HV$  を達成したと考えられる.



(a) 実行可能率  $\phi = 0.1$       (b) 実行可能率  $\phi = 0.3$       (c) 実行可能率  $\phi = 0.5$

図 6.16: 適応制御した  $S$  の推移 ( $m = 2$  目的問題)

## 6.6 まとめ

本章では、まず、指向性交配による解探索性能をさらに高めるために、指向性交配における第二の親候補集合  $\mathcal{M}$  の目的空間における選出領域をパラメータ  $S$  によって拡大・縮小する TNSDM-CS を提案し、その解探索性能を離散問題の  $mk$ -KP と連続問題の mCDTLZ 問題を用い、TNSDM、従来法の CNSGA-II と比較した。その結果、 $mk$ -KP と mCDTLZ の両方の問題において、指向性交配における第二の親候補集合  $\mathcal{M}$  の選出領域を制御することにより、解探索性能が改善されることが確かめられた。解探索性能が改善される要因は、パラメータ  $S$  小さくした場合の指向性交配の実行回数の増加と、パラメータ  $S$  を大きくした場合の解探索の指向性の強化によることがわかった。そのため、双方バランスが良好な  $S^*$  を設定したときに、最も高い  $HV$  が達成されることがわかった。

その結果を基に、次に、一度の解探索中にパラメータ  $S$  を適応的に決定する TNSDM-ACS を提案した。 $mk$ -KP を用いた実験の結果、TNSDM-ACS は、有用な実行不可能解の選出領域を適応的に制御することによって、パラメータ  $S$  のチューニングを必要とせず、一度の解探索で選出領域制御法による解探索性能の改善効果を得ることが確かめられた。特に実行可能率が高い問題において、TNSDM-ACS は交配ごとに異なる  $S$  を適応的に決定することによって、最適な  $S^*$  を用いる TNSDM-CS より高い解探索性能を達成できることが確かめられた。

## 第 7 章

# TNSDM における有用な不可能解のアーカイブ法

### 7.1 指向性交配の問題 2 : 有用な実行不可能解の消失

指向性交配で選出される解集合  $M$  は, 制約付き多目的最適化を促進する有用な解といえる. しかし,  $M$  は実行不可能解であることが多いため, 解集団から淘汰されやすく, 次世代には残りにくい.  $M$  に選択される解探索に有用な実行不可能解をアーカイブして次世代に維持し, 繰り返し指向性交配する機会を与えることによって, さらに TNSDM の解探索性能を改善出来る可能性がある.

### 7.2 TNSDM-A : 有用な実行不可能解をアーカイブする TNSDM

指向性交配による解探索性能の改善効果をさらに高めるために, 有用な実行不可能解のアーカイブ法を構築し, TNSDM に組み込んだ TNSDM-A (TNSDM using Archive) を提案する. 図 7.1 に TNSDM-A のブロック図を示す. 基礎となる TNSDM に対して, ブロック図中の青色で示した処理を追加する. また, TNSDM-A の疑似コードを **Algorithm 3** に示す. また, 表 7.1 にブロック図と疑似コードで用いる記号を示す. **Algorithm 3** と表 7.1 では, 基礎となる TNSDM の記述を黒色で, TNSDM-A のための記述を青色で示す. TNSDM-A は親集団  $\mathcal{P}$  と子集団  $\mathcal{Q}$  からなる全解集団  $\mathcal{R} (= \mathcal{P} \cup \mathcal{Q})$  を用いる. さらに, 新たに指向性交配に有用な解を保持するためのアーカイブ集団  $\mathcal{A}$  を導入する.

まず, TNSDM-A では  $\mathcal{R} \cup \mathcal{A}$  のすべての解を二段階の非支配ソートによってフロント分類を行い, 上位解から親集団  $\mathcal{P}$  を選択する. その後, 指向性交配によって子を生成していく. 指向性交配では従来と同様に第一の親  $p_a$  を親集団  $\mathcal{P}$  からバイナリトーナメント

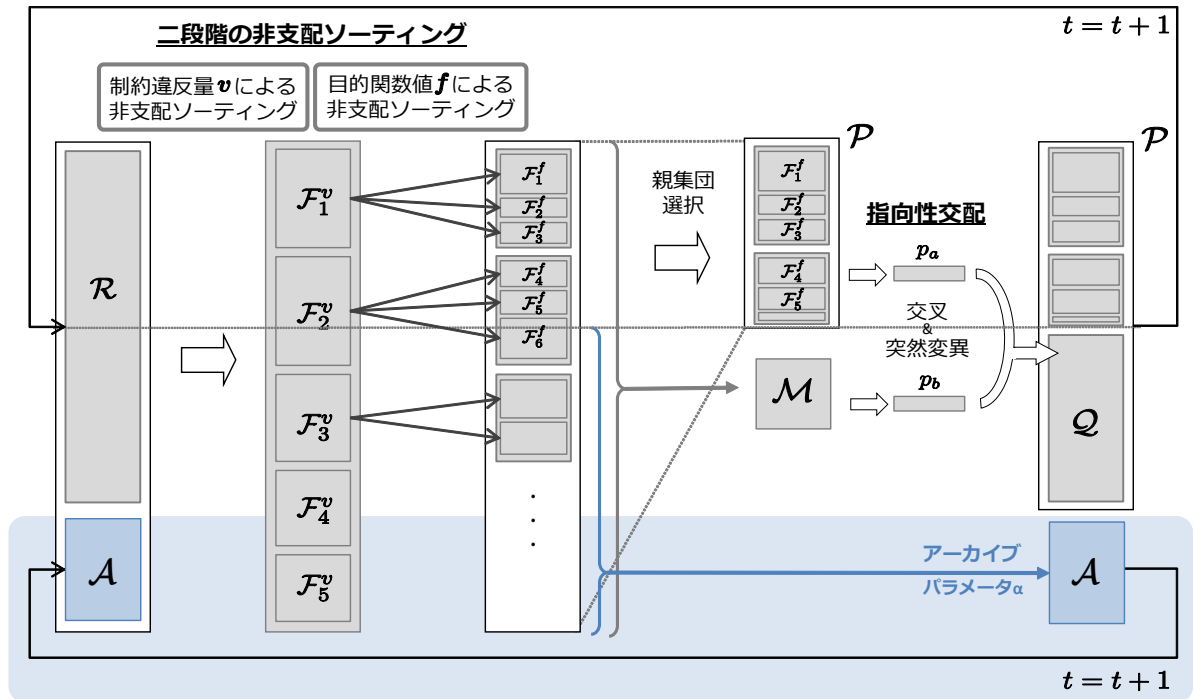


図 7.1: TNSDM-A のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団,  $\mathcal{A}$ : アーカイブ集団)

選択により選択する。そして,  $p_a$  を支配する第二の親候補集団  $\mathcal{M}$  を  $\mathcal{R} \cup \mathcal{A}$  から選出する ( $\mathcal{M} = \{x \in \mathcal{R} \cup \mathcal{A} \mid x \succ_f p_a\}$ )。ここで,  $\mathcal{M}$  に含まれる解は指向性交配に有用な解であるため,  $\mathcal{M}$  中の上位フロントに属す順に  $\alpha$  個体にマークを付ける。  $\alpha$  は, TNSDM-A のパラメータである。この際, 同一フロントの解の優劣は, CD によって決定する。なお, このマークはアーカイブ集団  $\mathcal{A}$  の選択のために用いられる。次に, TNSDM と同様に,  $\mathcal{M}$  から第二の親  $p_b$  を選択し, 子を生成する。これを繰り返す, 子集団  $\mathcal{Q}$  を生成する。

次に, アーカイブ集団  $\mathcal{A}$  を選択する。  $\mathcal{R} \cup \mathcal{A}$  から, マークのついた解でかつ親集団  $\mathcal{P}$  に選択されていない解を新たなアーカイブ集団  $\mathcal{A}$  として選択する (Algorithm 3 23 行目)。すなわち, アーカイブ集団  $\mathcal{A}$  は, 親集団でありエリート解集団である  $\mathcal{P}$  からは排除された解の中で指向性交配に有用な解としてマークされた解集団である。また, 次世代のアーカイブ集団  $\mathcal{A}$  は, 現世代の解集団  $\mathcal{R} \cup \mathcal{A}$  から選択される。そのため, 現世代のアーカイブ解が, 次世代でもアーカイブ解になることもあれば, より良い実行不可能解に置き換わることもある。このように, アーカイブ集団  $\mathcal{A}$  内の実行不可能解は, より制約違反量が小さく, より目的関数値の良い解へと進化する。また,  $\mathcal{M}$  に選出された解からアーカイブする理由は, 親集団内の解を支配できる目的関数値を有することと, 目的関数空間に広く分布するアーカイブ集団を保持するためである。

TNSDM-A では, パラメータ  $\alpha$  によってアーカイブサイズを変更することができる。

---

**Algorithm 3** TNSDM-A

---

```
1: Randomly generate initial solutions ( $\mathcal{R}$ )
2:  $\mathcal{A} = \emptyset$ 
3: for  $t = 1$  to  $T$  do
4:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
5:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sort}(\mathcal{R} \cup \mathcal{A}, \succ_v)$ 
6:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
7:      $j = 1 + |\mathcal{F}^f|$ 
8:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sort}(\mathcal{F}_i^v, \succ_f)$ 
9:   end for
10:  Crowding distance ( $\mathcal{F}^f$ )
11:   $\mathcal{P} = \text{Truncation}(\mathcal{R} \cup \mathcal{A}, |\mathcal{R}|/2)$ 
12:  for  $j = 1$  to  $|\mathcal{Q}|$  do
13:     $p_a = \text{Tournament selection}(\mathcal{P})$ 
14:     $\mathcal{M} = \{\mathbf{x} \in \mathcal{R} \cup \mathcal{A} \mid \mathbf{x} \succ_f p_a\}$ 
15:    Mark best  $\alpha$  solutions selected in  $\mathcal{M}$  ( $\{\mathbf{x} \in \mathcal{R} \cup \mathcal{A} \mid \mathbf{x} \succ_f p_a\}$ )
16:    if  $p_a$  is feasible or  $|\mathcal{M}| \geq 2$  then
17:       $p_b = \text{Tournament selection}(\mathcal{M}) : \textit{Directed Mating}$ 
18:    else
19:       $p_b = \text{Tournament selection}(\mathcal{P}) : \textit{Conventional Mating}$ 
20:    end if
21:     $Q_j = \text{Crossover and Mutation}(p_a, p_b)$ 
22:  end for
23:   $\mathcal{A} = \{\mathbf{x} \in \mathcal{R} \cup \mathcal{A} \mid \mathbf{x} \text{ is a marked solution} \wedge \mathbf{x} \notin \mathcal{P}\}$ 
24:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
25: end for
26:  $\mathcal{POS} = \{\mathbf{x} \in \mathcal{R}^{feasible} \mid \neg \exists \mathbf{y} \in \mathcal{R}^{feasible} : \mathbf{y} \succ_f \mathbf{x}\}$ 
```

---

また、 $\alpha = 0$  であるとき、TNSDM と等しい。  $\alpha$  の増加に伴ってアーカイブサイズが大きくなる。

### 7.3 実験内容

指向性交配に有用な解をアーカイブする効果を検証するために、TNSDM-A のアーカイブサイズパラメータ  $\alpha$  を変化させ、解探索性能を評価する。比較アルゴリズムとして、



表 7.1: 擬似コード内で使用される記号 (TNSDM-A)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{A}$	アーカイブ集団
$\alpha$	アーカイブサイズパラメータ
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

TNSDM(TNSDM-A における  $\alpha = 0$ ), 加えて CNSGA-II を用いる. また, テスト問題として離散問題の  $mk$ -KP, 連続問題の mCDTLZ を用いる.  $mk$ -KP では, 問題の目的数と制約の厳しさ (実行可能率), mCDTLZ では問題の目的数 (=制約数) を変化させた場合のアーカイブ効果について調査する.

### 7.3.1 パラメータ

$mk$ -KP の問題パラメータは, アイテム数  $n = 500$ , 目的数  $m = \{2, 4, 6\}$ , ナップサック (制約) 数  $k = 6$ , 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  に設定した.  $mk$ -KP におけるアルゴリズムの遺伝的パラメータは, 一様交叉法 (交叉率  $P_c = 1.0$ ), ビット反転突然変異法 (突然変異率  $P_m = 1/n$ ), 世代数  $T = 10,000$  を用いた. mCDTLZ の問題パラメータは, 目的数 (=制約数)  $m = \{2, 4, 6\}$ , 変数の数  $n = 12$  とした. mCDTLZ におけるアルゴリズムの遺伝的パラメータは, SBX 交叉法 (交叉率  $P_c = 0.8$ , 分布パラメータ  $\eta_c = 15$ ), Polynomial mutation (突然変異率  $P_m = 0.2$ , 分布パラメータ  $\eta_m = 20$ ), 世代数は  $T = 5,000$  を用いる. 集団サイズは  $mk$ -KP, mCDTLZ いずれも  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) とした. ここで, アーカイブ集団  $\mathcal{A}$  を加える TNSDM-A は,  $\alpha$  の増加に伴って比較アルゴリズムよ

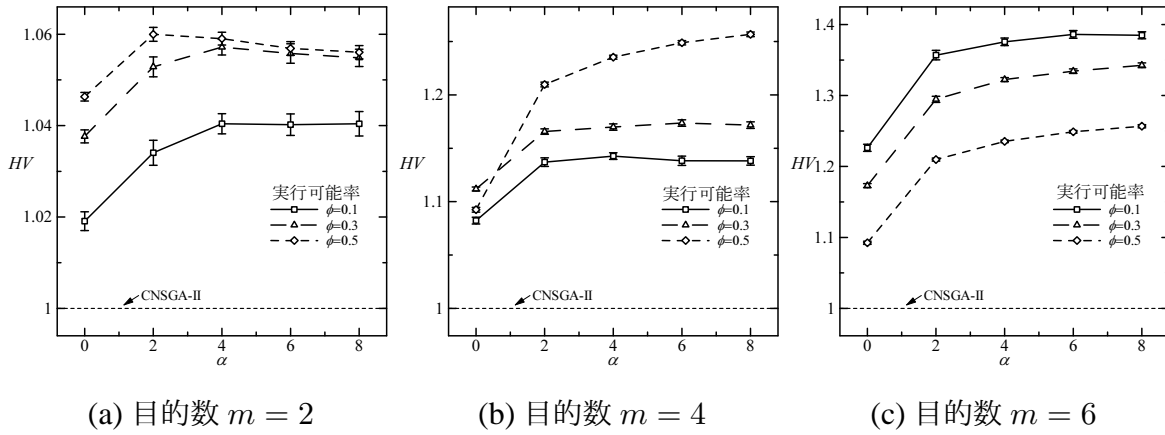


図 7.2:  $mk$ -KP における最終世代の  $HV$

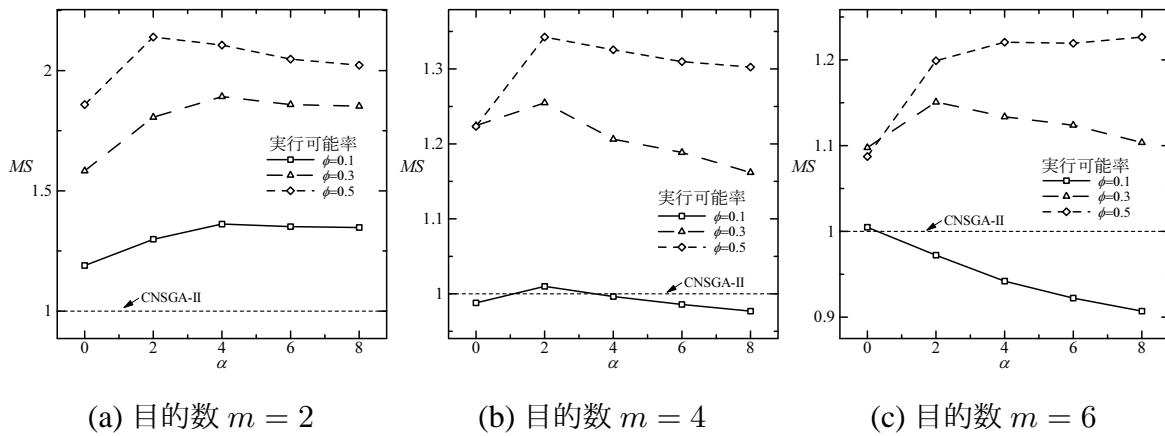


図 7.3:  $mk$ -KP における最終世代の  $MS$

りも解集団中に多くの解を保持するが、解の評価回数 (新たな解探索点となる子の生成回数) が同じという条件のもと、単純にアーカイブ集団を加える効果を検証する。実験の結果は 50 試行の平均値を用いて示す。

### 7.3.2 評価尺度

獲得した解集合を総合評価する Hypervolume ( $HV$ ) によって TNSDM-A の効果を検証する。 $HV$  の算出において、参照点は  $mk$ -KP では  $\mathbf{r} = (0, 0, \dots, 0)$ , mCDTLZ では  $\mathbf{r} = (1.1, 1.1, \dots, 1.1)$  を用いた。その後、多様性と収束性を、それぞれ Maximum Spread ( $MS$ ),  $C$  メトリックを用いて詳細に評価する。

## 7.4 実験結果と考察

### 7.4.1 $mk$ -KP における $HV$ の比較

まず、アーカイブ法を導入した TNSDM-A においてアーカイブサイズ  $\alpha$  を変化させた場合の解探索性能を  $mk$ -KP を用いて検証する。図 7.2 に最終世代で得られた解集合 POS の  $HV$  の結果を示す。また、エラーバーは 95% 信頼区間を示している。 $\alpha = 0$  とした TNSDM-A は TNSDM に等しく、 $\alpha > 0$  とすることによってアーカイブ機構が働く。また、従来法の CNSGA-II の結果もプロットしており、すべてアルゴリズムの結果は CNSGA-II より得られた結果を 1.0 として正規化している。

これらの結果から、まず、すべての問題で TNSDM が従来法の CNSGA-II よりも高い  $HV$  を示すことがわかる。さらに、TNSDM ( $\alpha = 0$ ) よりもアーカイブ法を導入した TNSDM-A ( $\alpha > 0$ ) が高い  $HV$  を示している。よって、TNSDM は指向性交配に有用な解のアーカイブ法を導入することによって解探索性能が向上することが明らかになった。また、TNSDM-A は解く問題によって  $HV$  を最大化させる最適なパラメータ  $\alpha^*$  があることがわかる。 $m = 2$  目的の場合、実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の問題に対して、それぞれ最適パラメータは  $\alpha^* = \{4, 4, 2\}$  となっている。低い実行可能率  $\phi$  の問題ほど、指向性交配において第二の親候補解集合  $\mathcal{M}$  に解が含まれやすい一方で最適なアーカイブサイズ  $\alpha^*$  が大きくなっている。これは、 $\mathcal{M}$  内でより解ランクが高い有用なアーカイブ解の選択圧を高めるためと考えられる。 $m = \{4, 6\}$  目的では実行可能率の上昇に伴って、最適なアーカイブサイズ  $\alpha^*$  が大きくなる。これは、実行可能率の高い問題では解集団中に実行可能解が減少するのに加えて、目的数の増加により解同士の支配による優劣が付きづらくなり、第二の親候補解集合  $\mathcal{M}$  が得られにくくなるため、アーカイブ解を多数保持することで、指向性交配を実行しやすくするためと考えられる。ただし、目的空間における解の分布は均一でないため、各親  $p_a$  に対して最適なアーカイブサイズ  $\alpha^*$  が異なる可能性がある。

### 7.4.2 $mk$ -KP における POS の多様性と収束性

得られた最適解集合について、目的空間における広がりを表す多様性と真の最適パレートフロントへの収束性を、それぞれ  $MS$  と  $C$  メトリックを用いて分析する。

まず、多様性の結果を図 7.3 に示す。 $MS$  の結果は  $HV$  と同様に CNSGA-II より得られた値を 1.0 として正規化してプロットしている。 $m = 6$  目的、実行可能率  $\phi = 0.1$  の場合を除いて、アーカイブを加えることによって、TNSDM よりも  $MS$  を改善するパラメータ  $\alpha$  が存在することがわかる。また、 $m = 4$  目的、実行可能率  $\phi = 0.3$  問題と  $m = 6$  目

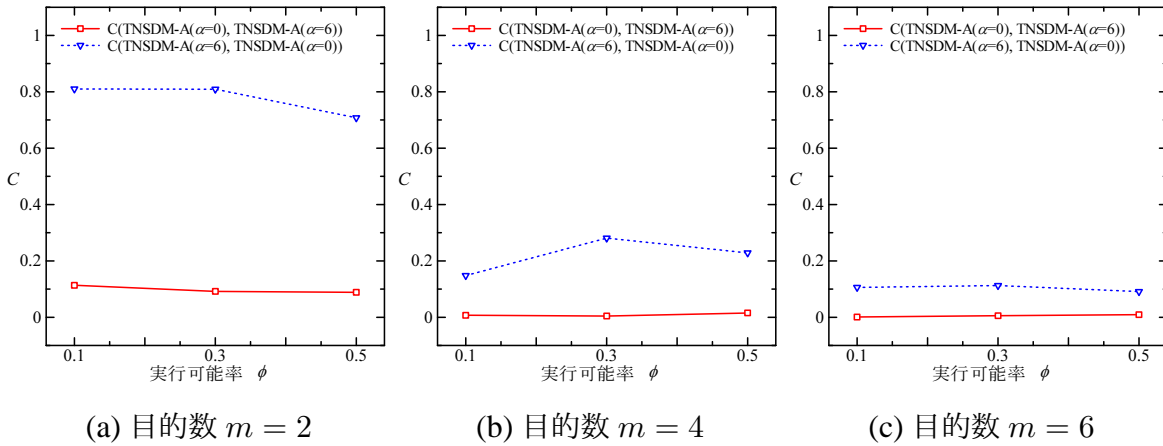


図 7.4:  $mk$ -KP における  $C$  メトリックによる比較

的, 実行可能率  $\phi = 0.1$  問題を除いて,  $HV$  を最も高める  $\alpha$  において, TNSDM よりも高い  $MS$  を示し, アーカイブを加えることによって多様性が向上することがわかる.

次に, 収束性について  $C$  メトリックを用いて評価する.  $C$  メトリックでは, ふたつのアルゴリズム間の結果しか出すことができないため, TNSDM-A( $\alpha = 0$ ) と TNSDM-A( $\alpha = 6$ ) で獲得した POS を用いてアーカイブを加えた場合の収束性を検証する. 図 7.4 に結果を示す. これらの結果から,  $C(\text{TNSDM-A}(\alpha = 6), \text{TNSDM-A}(\alpha = 0))$  の値が  $C(\text{TNSDM-A}(\alpha = 0), \text{TNSDM-A}(\alpha = 6))$  よりもすべての問題で値が上回っていることがわかる. すなわち, TNSDM-A( $\alpha = 6$ ) で獲得した POS のうちより多くの解が TNSDM-A( $\alpha = 0$ ) で獲得した POS の解を支配していることを示している. このことから, アーカイブ法を加えることによって, POS の収束性が高まったことがわかる.

$MS$  と  $C$  メトリックの結果から, TNSDM に指向性交配に有用な解のアーカイブ法を加えることによって, 獲得する POS は,  $m = 4$  目的, 実行可能率  $\phi = 0.3$  問題と  $m = 6$  目的, 実行可能率  $\phi = 0.1$  問題では収束性が, その他の問題では多様性と収束性の両面が向上することがわかった.

### 7.4.3 mCDTLZ における $HV$ の比較

次に, mCDTLZ における TNSDM-A の解探索性能を  $HV$  を用いて検証する. 図 7.5 に最終世代で得られた解集合の  $HV$  の結果を示す. すべてのアルゴリズムの結果は CNSGA-II より得られた結果を 1.0 として正規化している. また, エラーバーは 95% 信頼区間を示している.

これらの結果から, TNSDM が  $m = 6$  目的問題で CNSGA-II と同等,  $m = \{2, 4\}$  目的問題で CNSGA-II よりも高い  $HV$  を示すことがわかる. さらに, TNSDM ( $\alpha = 0$ ) よりもアーカイブ法を導入した TNSDM-A( $\alpha > 0$ ) が高い  $HV$  を示している. よって,

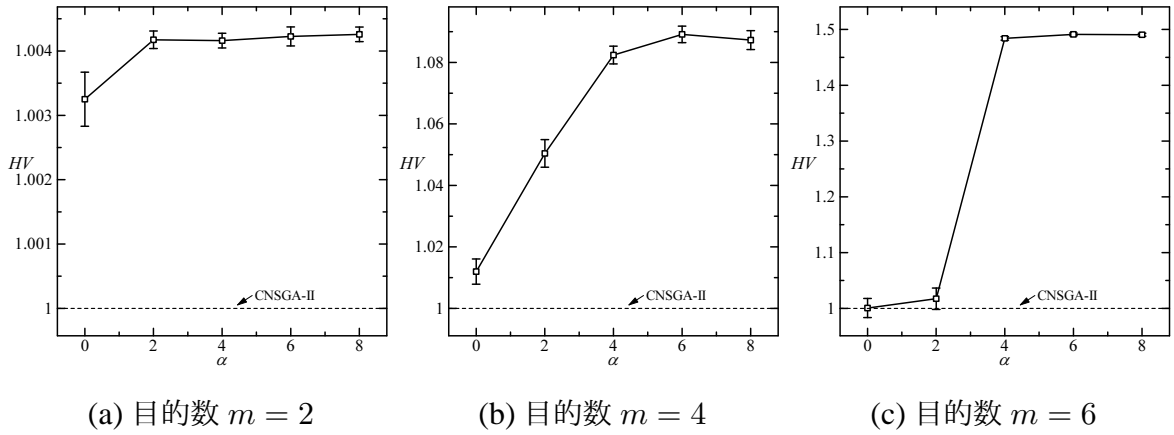


図 7.5: mCDTLZ における最終世代の  $HV$

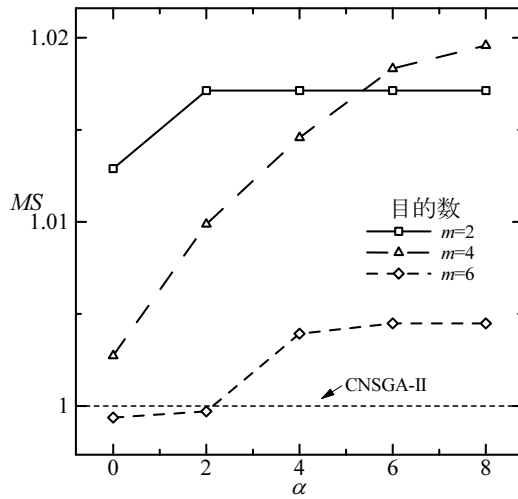


図 7.6: mCDTLZ における最終世代の  $MS$

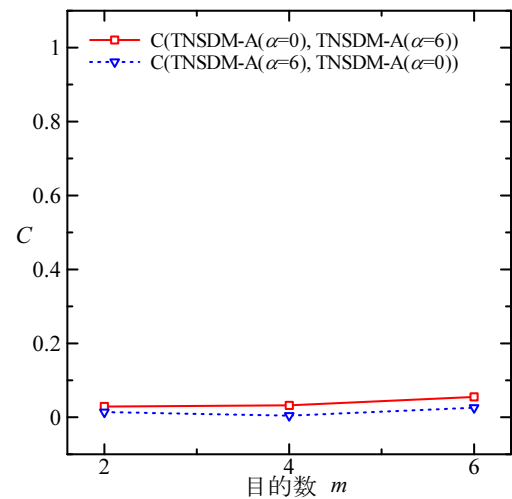


図 7.7: mCDTLZ における  $C$  メトリックによる比較

mCDTLZ においても, TNSDM は指向性交配に有用な解のアーカイブ法を導入することによって解探索性能が向上することが明らかになった.

#### 7.4.4 mCDTLZ における POS の多様性と収束性

得られた最適解集合について, 目的空間における広がりを表す多様性と真の最適パレートフロントへの収束性を, それぞれ  $MS$  と  $C$  メトリックを用いて分析する.

まず, 多様性の結果を図 7.6 に示す. なお,  $MS$  の値はすべて CNSGA-II より得られた値を 1.0 として正規化している. これらの結果から,  $MS$  の結果は  $HV$  の結果と同様な傾向があることがわかる. TNSDM(TNSDM-A,  $\alpha = 0$ ) は,  $m = 6$  目的問題を除いて,

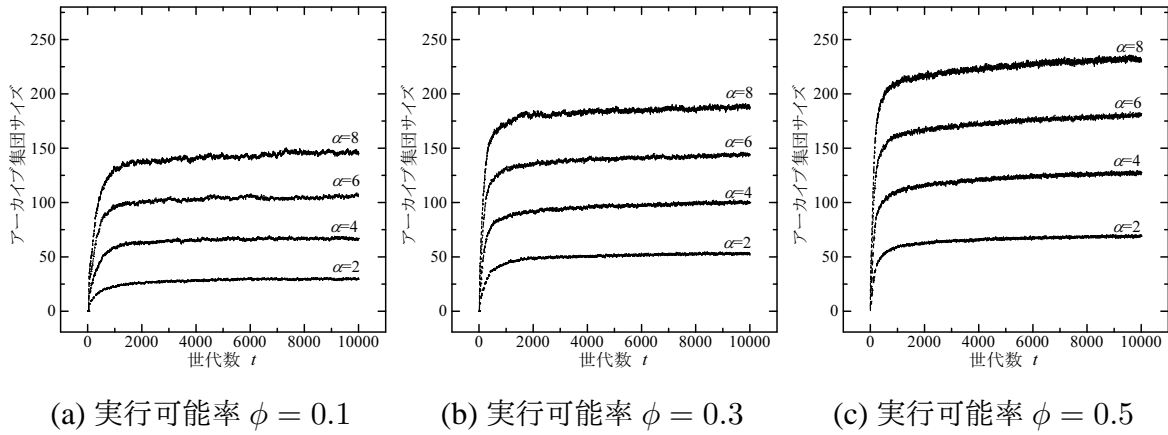


図 7.8:  $mk$ -KP におけるアーカイブ集団サイズ  $|A|$  の世代推移 ( $m = 2$  目的)

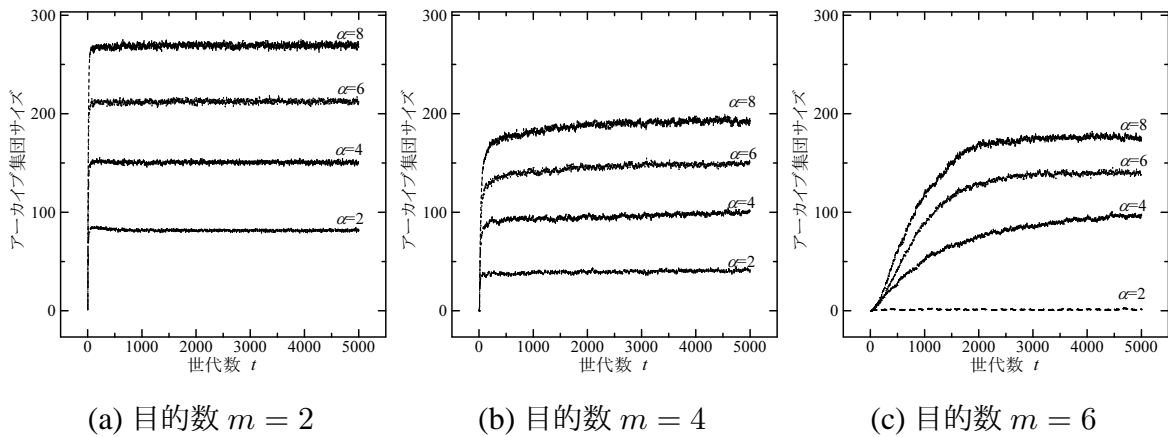


図 7.9: mCDTLZ におけるアーカイブ集団サイズ  $|A|$  の世代推移

従来法の CNSGA-II よりも高い  $MS$  を示しており、CNSGA-II と比較して獲得した POS の多様性が高いことがわかる。次にアーカイブ法の効果について、アーカイブを加えることによって ( $\alpha > 0$ )、 $MS$  値が上昇し POS の多様性が向上することがわかる。

次に、収束性について  $C$  メトリックを用いて評価する。 $C$  メトリックでは、ふたつのアルゴリズム間の結果しか出すことができないため、TNSDM-A( $\alpha = 0$ ) と TNSDM-A( $\alpha = 6$ ) で獲得した POS を用いてアーカイブを加えた場合の収束性を検証する。図 7.7 に結果を示す。これらの結果から、 $C(\text{TNSDM-A}(\alpha = 6), \text{TNSDM-A}(\alpha = 0))$  の値は  $C(\text{TNSDM-A}(\alpha = 0), \text{TNSDM-A}(\alpha = 6))$  よりもすべての問題で値が下回っていることがわかる。すなわち、TNSDM-A( $\alpha = 6$ ) で獲得した POS は TNSDM-A( $\alpha = 0$ ) で獲得した POS よりも収束の点で劣ることがわかる。

これらの結果から、mCDTLZ においてアーカイブ法を加えることによって、特に多様性の向上によって解探索性能が向上したことがわかった。

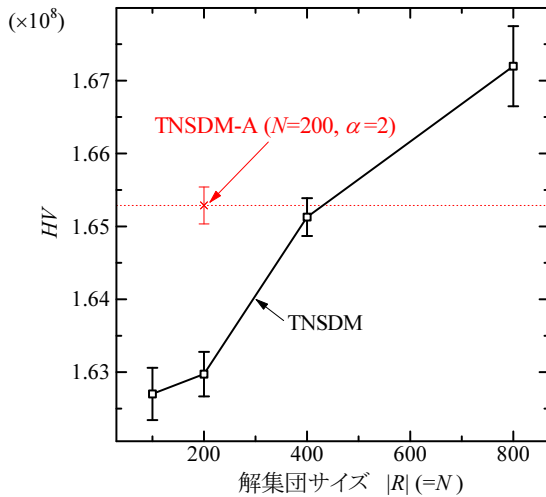


図 7.10: 解集団サイズの変化に対する  $HV$

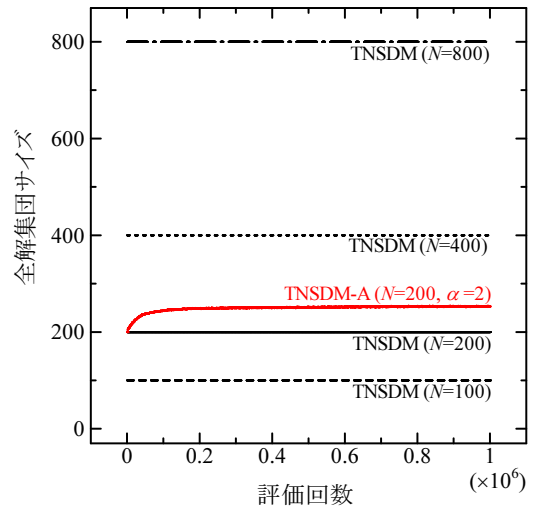


図 7.11: 取り扱う解集団サイズ

#### 7.4.5 アーカイブサイズ $\alpha$ とアーカイブ集団サイズ $|\mathcal{A}|$ の関係

$m = 2$  目的  $k = 6$  制約, 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の  $mk$ -KP と,  $m = \{2, 4, 6\}$  目的 (=制約) の mCDTLZ において, アーカイブサイズ  $\alpha$  を変化させた場合のアーカイブ集団サイズ  $|\mathcal{A}|$  の世代推移をそれぞれ図 7.8 と図 7.9 に示す. アーカイブ集団サイズ  $|\mathcal{A}|$  は  $\alpha$  の増加に伴って大きくなるのがわかる. また, 毎世代ごと, 問題ごとに異なるのがわかる. TNSDM-A におけるアーカイブパラメータ  $\alpha$  は, それぞれの親  $p_a$  に対してアーカイブする解の個数であるため, 毎世代におけるアーカイブ集団サイズ  $|\mathcal{A}|$  の最大は,  $\alpha \times |\mathcal{P}|$  (本実験では  $|\mathcal{P}| = 100$ ) となる. しかし, 図 7.8 と図 7.9 に示されるように, 実際のアーカイブ集団サイズ  $|\mathcal{A}|$  は最大値よりも小さくなっている. その要因となるのは, 指向性交配の回数と, ひとつの解が複数の親  $p_a$  からアーカイブの対象として指定されることである.

#### 7.4.6 アーカイブ集団の保持と解集団サイズの増加

次に, TNSDM-A においてアーカイブ集団を用いる場合と TNSDM において解集団サイズを単純に増加した場合の解探索性能を検証する.  $mk$ -KP ( $m = 2$  目的,  $k = 6$  制約, 実行可能率  $\phi = 0.3$ ) において, 解集団サイズ  $N = \{100, 200, 400, 800\}$  を用いる TNSDM と,  $N = 200$  と  $\alpha = 2$  を用いる TNSDM-A を比較する. ここで, 異なる解集団サイズ  $N$  で同じ世代数進化させた場合, 新たな解探索点となる子個体生成数が異なるため,  $10^6$  回評価 (子個体生成) 後の  $HV$  によって性能を評価する. 結果を図 7.10 に示す. この結果から, まず, アーカイブ集団を用いない TNSDM は, 解集団サイズ  $N$  の増

加に伴って、 $HV$  が改善することがわかる。次に、アーカイブ集団と  $N = 200$  の解集団を用いる TNSDM-A は、 $N = 200$  を用いる TNSDM より高い  $HV$  を示し、さらに、その 2 倍の解集団  $N = 400$  を用いる TNSDM よりも高い  $HV$  を示すことがわかる。次に、解探索中に取り扱う解集団  $\mathcal{R} \cup \mathcal{A}$  のサイズの推移を図 7.11 に示す。アーカイブ集団を用いない TNSDM が一定数の解集団  $N$  を取り扱うのに対して、アーカイブ集団を用いる TNSDM-A は取り扱う解集団サイズが動的に変化する。 $N = 200$  を用いる TNSDM-A は、アーカイブ集団を保持するため、取り扱う解集団サイズは 200 以上になるが、400 以下であることがわかる。図 7.10 と図 7.11 の結果から、TNSDM-A によって実行不可能解をアーカイブして解探索に活用することは、単純に解集団サイズ  $N$  を増やすことより解探索性能を高める効果があることが明らかになった。

## 7.5 まとめ

TNSDM において、指向性交配の効果をさらに高めるため、これまで世代ごとに解集団から消失していた有用な実行可能解を解集団中にアーカイブし、繰り返し親に選択して解探索に活用する TNSDM-A を提案した。テスト問題として  $mk$ -KP と mCDTLZ を用いて、アーカイブサイズ  $\alpha$  を変化させた場合の TNSDM-A によるアーカイブ法の効果を、TNSDM、従来法の CNSGA-II と比較し検証した。実験の結果、TNSDM-A が最も高い  $HV$  を示し、アーカイブ法によって解探索性能が向上することがわかった。



## 第 8 章

# 有用な実行不可能解の選出領域制御法とアーカイブ法の組み合わせ

### 8.1 選出領域制御法とアーカイブ法を組み合わせる意義

6 章では，有用な実行不可能解の選出領域を縮小すると，解探索の指向性が高まるため解探索性能が高まる利点があるが，同時に，選出領域に解が存在せずに指向性交配を実行できないケースが生じる欠点があることがわかった．この問題に対して，7 章で導入した有用な実行不可能解のアーカイブを導入すると，選出領域を縮小しても選出領域に解が得られやすくなるため，選出領域を縮小する欠点を補える．すなわち，アーカイブによって指向性交配を実行できないケースを軽減した上で，選出領域を縮小して解探索の指向性を高めた解探索が可能になる．このように，有用な実行不可能解の選出領域制御法のみ，アーカイブ法のみでは，得られない効果が，これらを組み合わせることによって実現できる可能性がある．

### 8.2 TNSDM-CSA：選出領域制御法とアーカイブ法を用いる TNSDM

本章では，TNSDM に有用な実行不可能解の選出領域制御法とアーカイブ法を組み合わせた TNSDM-CSA を提案する．図 8.1 に TNSDM-CSA のブロック図を示す．また，TNSDM-CSA の疑似コードを **Algorithm 4** に示す．ブロック図と疑似コード中では，基礎となる TNSDM に対して，有用な実行不可能解の選出領域制御の処理を赤色で，アーカイブに関する処理を青色で示す．また，表 8.1 にブロック図と疑似コードで用いる記号を示す．

TNSDM-CSA のアルゴリズムは，アーカイブ集団のみを用いる TNSDM-A において，

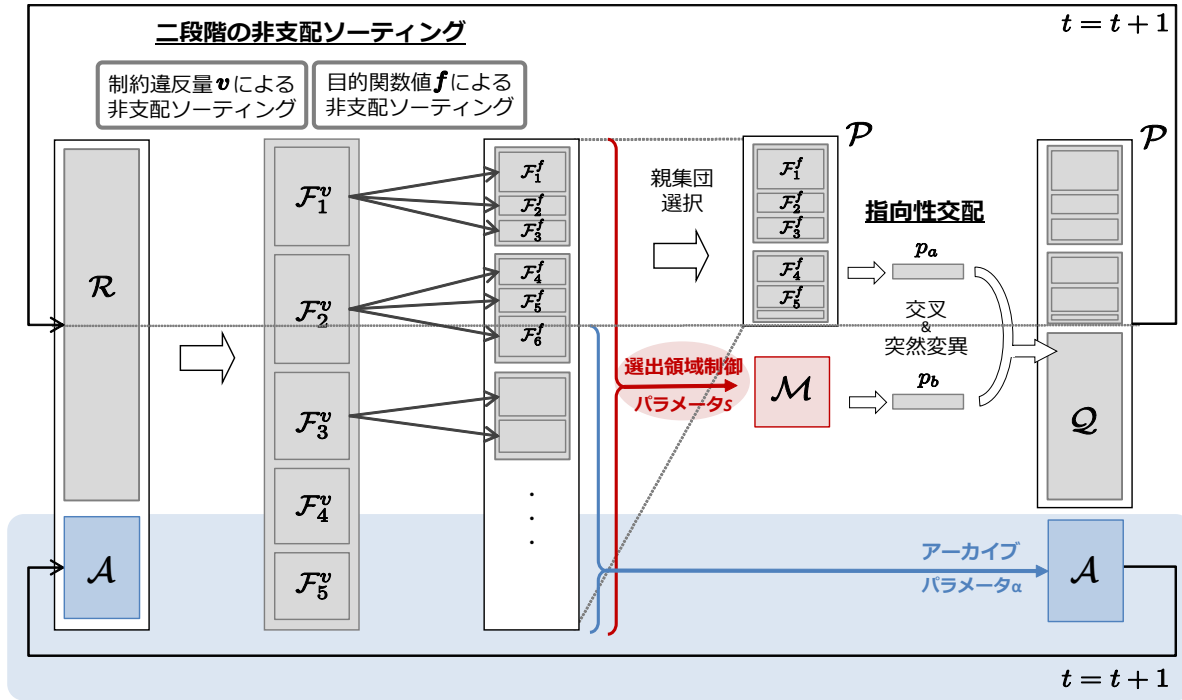


図 8.1: TNSDM-CSA のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団,  $\mathcal{A}$ : アーカイブ集団))

第二の親の候補解集合  $\mathcal{M}$  を選出するときに、式 (6.1) によって目的関数値を変換してから求める解の支配  $\succ_{f'}$  を用いる方法といえる。そのため、TNSDM-A が従来の解の支配  $\succ_f$  を使用するのに対し、本章で提案する TNSDM-CSA は、パラメータ  $S$  によって領域を制御した解の支配  $\succ_{f'}$  を用いる点異なる。また、TNSDM-CS は、第二の親の候補解集合  $\mathcal{M}$  を全解集団  $\mathcal{R}$  から選択するのに対し、本章で提案する TNSDM-CSA は、全解集団  $\mathcal{R}$  とアーカイブ集団  $\mathcal{A}$  から選択する点異なる。

TNSDM-CSA は、2つのパラメータを必要とする。一つ目は、有用な実行不可能解の選出領域を制御するためのパラメータ  $S$  であり、二つ目は、アーカイブのサイズのパラメータ  $\alpha$  である。

### 8.3 実験内容

$mk$ -KP を用いて、実行不可能解の選出領域制御とアーカイブを用いる TNSDM (TNSDM-CSA) において、選出領域制御のパラメータ  $S$  とアーカイブサイズのパラメータ  $\alpha$  を変化させた場合の指向性交配の実行回数、解探索性能について検証する。

---

**Algorithm 4** TNSDM-CSA

---

```
1: Randomly generate initial solutions ( $\mathcal{R}$ )
2:  $\mathcal{A} = \emptyset$ 
3: for  $t = 1$  to  $T$  do
4:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
5:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sort } (\mathcal{R} \cup \mathcal{A}, \succ_v)$ 
6:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
7:      $j = 1 + |\mathcal{F}^f|$ 
8:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sort } (\mathcal{F}_i^v, \succ_f)$ 
9:   end for
10:  Crowding distance ( $\mathcal{F}^f$ )
11:   $\mathcal{P} = \text{Truncation } (\mathcal{R} \cup \mathcal{A}, |\mathcal{R}|/2)$ 
12:  Calculate  $f'$  from  $f$  ( $\mathcal{R} \cup \mathcal{A}, S$ )
13:  for  $j = 1$  to  $|\mathcal{Q}|$  do
14:     $p_a = \text{Tournament selection } (\mathcal{P})$ 
15:     $\mathcal{M} = \{x \in \mathcal{R} \cup \mathcal{A} \mid x \succ_{f'} p_a\}$ 
16:    Mark best  $\alpha$  solutions selected in  $\mathcal{M}$  ( $\{x \in \mathcal{R} \cup \mathcal{A} \mid x \succ_{f'} p_a\}$ )
17:    if  $p_a$  is feasible or  $|\mathcal{M}| \geq 2$  then
18:       $p_b = \text{Tournament selection } (\mathcal{M}) : \textit{Directed Mating}$ 
19:    else
20:       $p_b = \text{Tournament selection } (\mathcal{P}) : \textit{Conventional Mating}$ 
21:    end if
22:     $Q_j = \text{Crossover and Mutation } (p_a, p_b)$ 
23:  end for
24:   $\mathcal{A} = \{x \in \mathcal{R} \cup \mathcal{A} \mid x \text{ is a marked solution} \wedge x \notin \mathcal{P}\}$ 
25:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
26: end for
27:  $\mathcal{POS} = \{x \in \mathcal{R}^{feasible} \mid \neg \exists y \in \mathcal{R}^{feasible} : y \succ_f x\}$ 
```

---

### 8.3.1 パラメータ

$mk$ -KP の問題パラメータは、アイテム数  $n = 500$ 、目的数  $m = \{2, 4, 6\}$ 、制約数  $k = 6$ 、実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  とした。遺伝的パラメータとして、解集団サイズは、 $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ )、世代数  $T = 10,000$  世代、一様交叉 (交叉率  $P_c = 1.0$ )、

表 8.1: 擬似コード内で使用される記号 (TNSDM-CSA)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{A}$	有用な実行不可能解のアーカイブ集団
$\alpha$	アーカイブサイズパラメータ
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$\mathcal{S}$	第二の親の候補解集合 $\mathcal{M}$ の選出領域を制御するパラメータ
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

ビット反転突然変異法 (変異率  $P_m = 1/n$ ) を用いた. TNSDM-CSA の計算コストは, アーカイブのパラメータ  $\alpha$  によって増加するが, 本章ではすべてのアルゴリズムを同じ世代数  $T$  (同じ解の評価回数) のもとで比較することとする. 結果は, 各アルゴリズムの 50 回の試行の平均値を用いて比較する.

### 8.3.2 評価尺度

獲得した解集合を総合評価する Hypervolume( $HV$ ) によって, TNSDM-CSA の性能を検証する.  $HV$  を算出する参照点  $\mathbf{r}$  は, 目的空間の原点  $\mathbf{r} = (0, 0, \dots, 0)$  に設定した.

## 8.4 実験結果と考察

### 8.4.1 指向性交配の実行回数

TNSDM-CSA の解探索性能を検証する前に, まず, TNSDM-CSA による解探索中における指向性交配の実行回数について調べる.  $m = \{2, 4, 6\}$  目的と実行可能率  $\phi =$

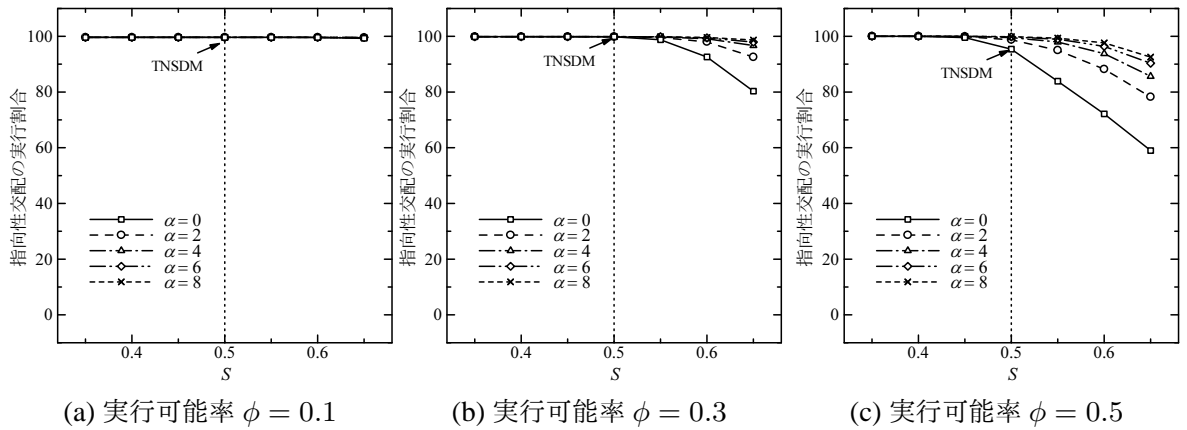


図 8.2: 指向性交配の実行割合 ( $m = 2$  目的問題)

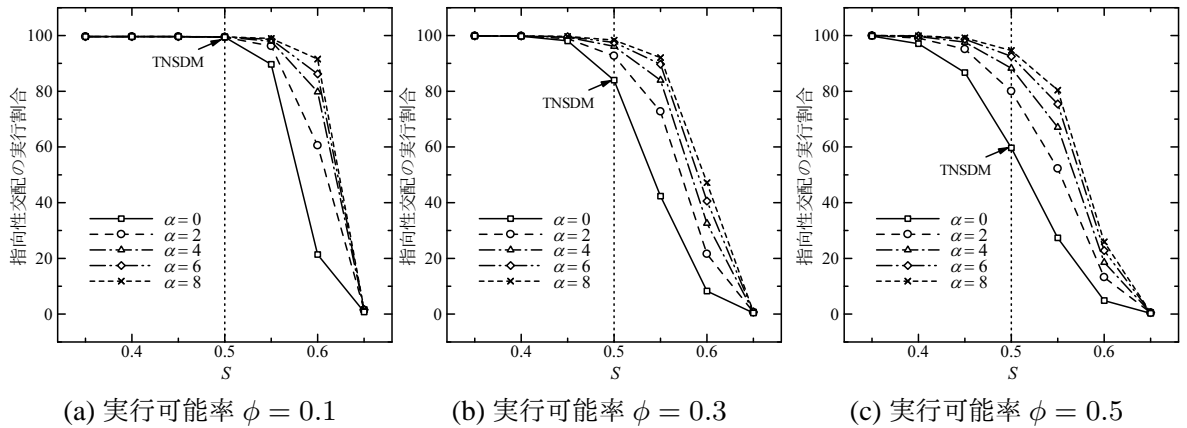


図 8.3: 指向性交配の実行割合 ( $m = 4$  目的問題)

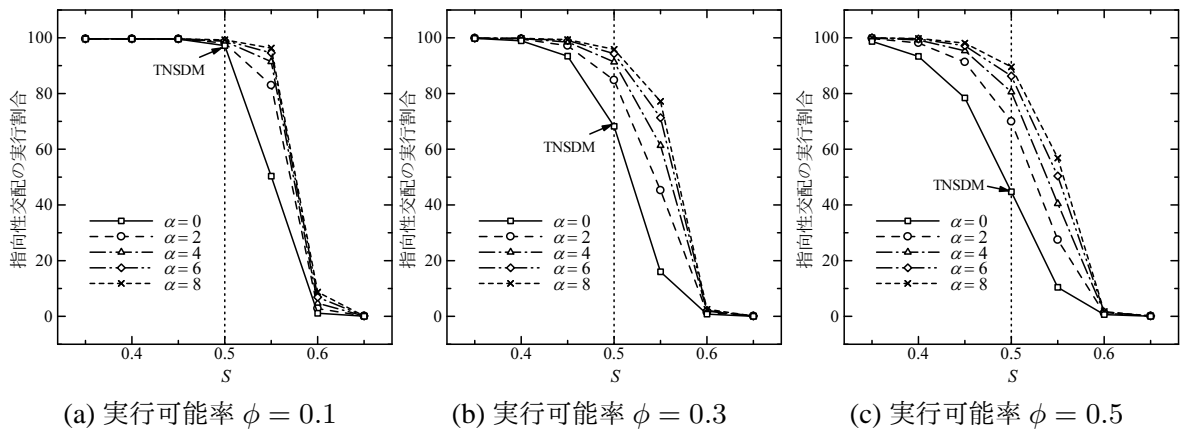


図 8.4: 指向性交配の実行割合 ( $m = 6$  目的問題)

$\{0.1, 0.3, 0.5\}$  の  $mk$ -KP において、TNSDM-CSA による解探索中のすべての交配の中で、指向性交配が実行された回数の割合を図 8.2~8.4 に示す。それぞれの図において、TNSDM-CSA の 2 つのパラメータ  $S$  と  $\alpha$  を変化させた場合の結果をプロットした。有用な実行不可能解  $\mathcal{M}$  の選出領域は、 $S$  を 0.5 より小さくすることによって拡大する。逆に、有用な実行不可能解  $\mathcal{M}$  の選出領域は、 $S$  を 0.5 より大きくすることによって縮小す

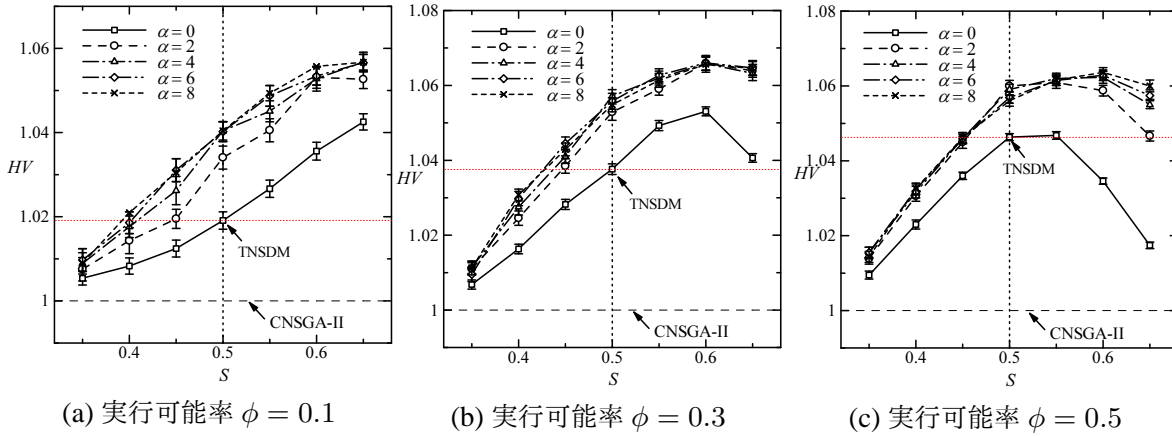


図 8.5: 最終世代の HV ( $m = 2$  目的問題)

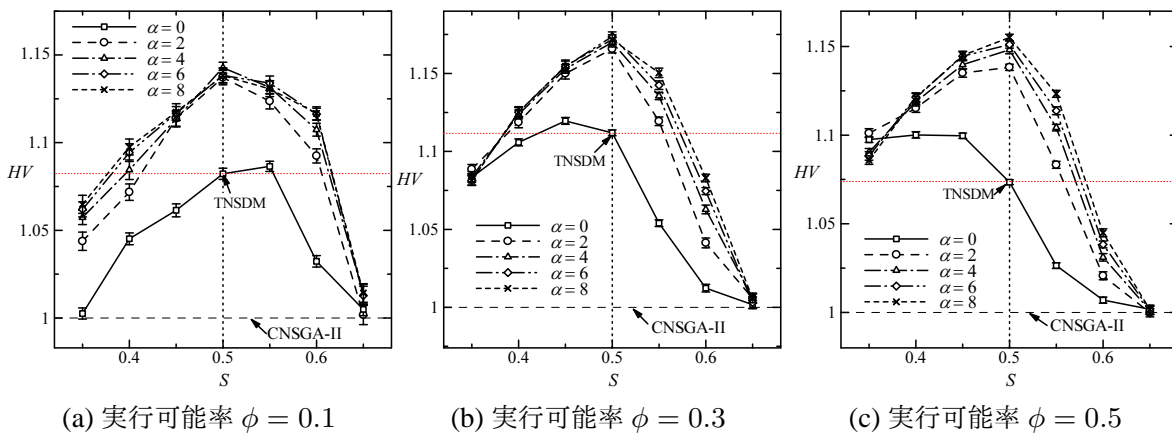


図 8.6: 最終世代の HV ( $m = 4$  目的問題)

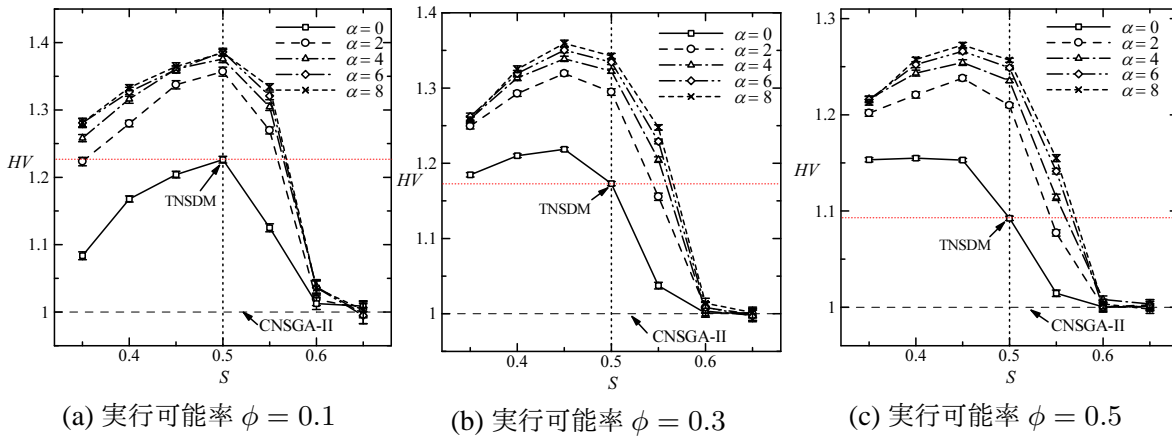


図 8.7: 最終世代の HV ( $m = 6$  目的問題)

る。また、TNSDM-CSA のアーカイブ集団のサイズは、 $\alpha$  の増加に伴って大きくなる。 $S = 0.5$  と  $\alpha = 0$  を設定した TNSDM-CSA は、5 章で提案した TNSDM と等しく、従来の選出領域を用いてアーカイブ集団を利用しない。 $\alpha = 0$  に設定した TNSDM-CSA は、6 章で提案した TNSDM-CS と等しい。 $S = 0.5$  に設定した TNSDM-CSA は、7 章で提案した TNSDM-A と等しい。

まず、TNSDM-CSA において  $S$  を変化させた場合の選出領域制御法の効果について述べる。図 8.2~8.4 の一般的な傾向として、パラメータ  $S$  の増加にともなって、指向性交配の実行回数が減少することがわかる。これは、有用な実行不可能解  $M$  の選出領域が縮小することによって、指向性交配を実行できなくなる第一の親の数が増加することが原因である。逆に、パラメータ  $S$  を減少させて有用な実行不可能解  $M$  の選出領域を拡大することによって、指向性交配を実行できる第一の親の数が増加することがわかる。また、 $mk$ -KP の実行可能率  $\phi$  の増加に伴って、指向性交配の実行回数が減少することがわかる。これは、問題の実行可能率  $\phi$  の増加に伴って、解集団中に生じる実行不可能解の数が減少することが原因である。しかし、実行可能率  $\phi$  が高い問題においても、 $S < 0.5$  を設定して有用な実行不可能解  $M$  の選出領域を拡大することによって、指向性交配の実行回数が増加することがわかる。

次に、TNSDM-CSA において  $\alpha$  を変化させた有用な実行不可能解のアーカイブ法の効果について述べる。図 8.2~8.4 の結果から、いずれの  $S$  を用いる TNSDM-CSA も、アーカイブのサイズ  $\alpha$  の増加に伴って、指向性交配の実行回数が増加することがわかる。すなわち、アーカイブ集団  $A$  に第二の親の候補解を維持することによって、指向性交配の実行回数が増加する。 $S$  の増加に伴って有用な実行不可能解  $M$  の選出領域が縮小するが、アーカイブ集団  $A$  の導入によって、指向性交配の実行回数を増加できることがわかった。

## 8.4.2 解探索性能

次に、TNSDM-CSA の解探索性能を検証する。 $m = \{2, 4, 6\}$  目的と実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の  $mk$ -KP における TNSDM-CSA の最終世代の  $HV$  の結果を図 8.5~8.7 に示す。それぞれの図において、TNSDM-CSA の 2 つのパラメータ  $S$  と  $\alpha$  を変化させた場合の結果をプロットした。また、エラーバーは 95% 信頼区間を示している。前述の通り、 $S = 0.5$  と  $\alpha = 0$  を設定した TNSDM-CSA は、5 章で提案した TNSDM と等しく、それぞれの図において、TNSDM の  $HV$  の結果を水平線で示した。また、従来の CNSGA-II の  $HV$  も結果も水平線でプロットした。ここで、すべての  $HV$  は CNSGA-II より得られた値を 1.0 として正規化している。

まず、TNSDM-CSA における有用な実行不可能解のアーカイブ法の効果について、アーカイブサイズのパラメータ  $\alpha$  を変化させた場合の  $HV$  について述べる。図 8.5~8.7 に示すすべての結果において、有用な実行不可能解のアーカイブを用いる TNSDM-CSA ( $\alpha > 0$ ) は、アーカイブを用いない TNSDM-CSA ( $\alpha = 0$ ) より高い  $HV$  を示すことがわかる。さらに、アーカイブのサイズ  $\alpha$  の増加に伴って、 $HV$  が改善することがわかる。この結果から、TNSDM-CSA では、有用な実行不可能解の選出領域制御のパラメータ  $S$  の値によらず、アーカイブ集団を用いることによって解探索性能が改善することがわかった。

次に、TNSDM-CSA における有用な実行不可能解の選出領域制御法の効果について、

横軸でパラメータ  $S$  を変化させた場合の  $HV$  について述べる．図 8.5～8.7 に示すいくつかの問題において，アーカイブサイズのパラメータ  $\alpha$  の増加に伴って，有用な実行不可能解の選出領域制御の最適なパラメータ  $S^*$  が大きくなる傾向が見て取れる．有用な実行不可能解の選出領域制御法では， $S$  を増加させると選出領域が縮小するため，第一の親の探索方向に近い  $M$  が選出され，解探索の指向性が高まる効果がある．しかし，目的空間における選出領域が縮小されるがゆえ，第二の親の候補解集合  $M$  に解が選出されにくくなり，指向性交配の実行回数が減少するとともに解探索性能も低下する傾向があった．これに対し，アーカイブ集団を導入すると，多数の有用な実行不可能解がアーカイブされるため，第二の親の候補解集合  $M$  が選出されずに指向性交配を実行できない恐れが軽減される．TNSDM-CSA では，有用な実行不可能解の選出領域制御法に加えてアーカイブ集団を利用できる．その結果， $S$  を増加させて選出領域を縮小して解探索の指向性を高めても，多数の有用な実行不可能解から第二の親を選択できるようになるため，解探索性能が改善したといえる．

このように，6 章と 7 章で提案した有用な実行不可能解の選出領域制御法とアーカイブ法を組み合わせることによって，指向性交配による解探索効果がさらに高まることが確かめられた．また，選出領域制御法とアーカイブ法を組み合わせた場合の  $HV$  の改善効果は，目的数の増加に伴って顕著になることもわかった．

## 8.5 まとめ

指向性交配による解探索効果をさらに高めることを目的として，6 章で提案した有用な実行不可能解の選出領域制御法と，7 章で提案した有用な実行不可能解のアーカイブ法を組み合わせた TNSDM-CSA を提案した． $mk$ -KP を用いた解探索性能の検証の結果，本章で提案した TNSDM-CSA は，選出領域制御法のみを用いる場合とアーカイブ法のみを用いる場合より，高い解探索性能を示すことが確かめられた．



## 第 9 章

# TNSDM における交叉量操作法

### 9.1 指向性交配の問題 3 : 遺伝子変異量

5~8 章で提案したアルゴリズムでは，指向性交配において，解探索に有用な実行不可能解の選択法にのみに注目しており，選択された実行可能解と実行不可能解の親には，従来から一般的に用いられる交叉法を適用してきた．具体的には，離散問題に対して，一様交叉法 [48]，連続問題に対して，SBX 交叉法 [49] を用いてきた．それぞれ，一様交叉法は遺伝子表現がバイナリ型，SBX 交叉法は変数型の場合に一般に用いられている．一様交叉法では，子は第一の親と第二の親の遺伝子を確率的に半分ずつ受け継ぐ．SBX 交叉法においても，二つの親の各遺伝子を変異させたのち，確率的に半分ずつの遺伝子を子に受け継ぐ．

指向性交配の場合，第一の親は実行可能解，第二の親は多くの場合実行不可能解となる．そのため，子は実行可能解から半分，実行不可能解から半分の遺伝子を受け継ぐこととなる．第二の親となる実行不可能解は，第一の親より目的関数値が良いことから，解探索を促進する有用な遺伝子情報を有する可能性がある．しかし，制約を違反しているため，その遺伝子を引き継ぐ子も実行不可能解になりやすい．図 9.1 に，2 目的 6 制約，実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の  $mk$ -KP を用いて，TNSDM において指向性交配を行う場合と指向性交配を行わず親集団から二つの親を選択する従来の交配法のみを用いた場合の，全世代で生成した子が実行可能解であった割合を百分率で示す．なお，実験に用いる遺伝的パラメータは，一様交叉法 (交叉率  $P_c = 1.0$ )，ビット反転突然変異法 (突然変異率  $P_m = 1/n$ )，世代数  $T = 10,000$ ，集団サイズ  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) とした．この結果から，実行不可能解を積極的に活用する指向性交配では，従来の交配法と比べて子が実行不可能解になりやすいことがわかる．最適解は実行可能解であるため，実行不可能解ばかり生成することは解探索に非効率的と考えられる．そのため，実行不可能解の親から子に遺伝する遺伝子量を少なくすることで実行可能解を生成しやすくなる可能性がある．解探索性能について， $HV$  の結果を図 9.2 に示す．ここでは，指向性交配を用いず従来の

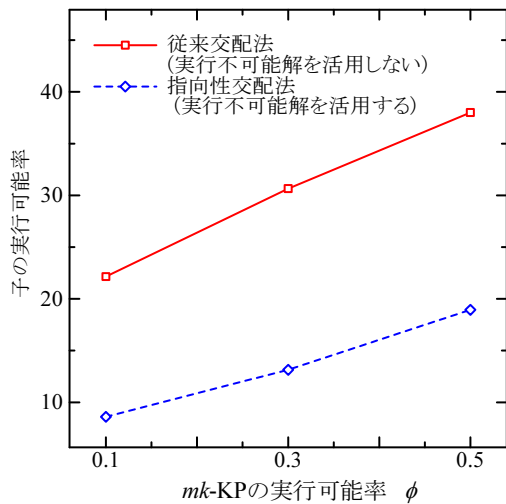


図 9.1: 指向性交配と従来の交配法による子の実行可能率

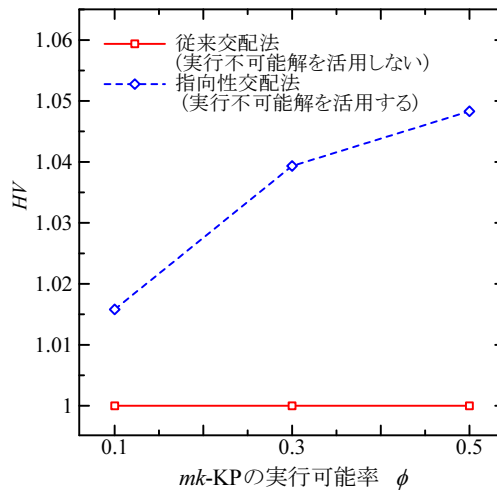


図 9.2: 指向性交配と従来の交配法による HV の結果

交配法のみを用いた TNSDM によって得られた結果を 1.0 として正規化してプロットしている。図 9.2 から、指向性交配は従来の交配より、解探索性能が高いことがわかる。実行不可能解を活用する指向性交配では、解が不可能解になりやすいが、従来の交配法と比べて生成された実行可能解が良質であるといえる。そのため、より多くの実行不可能解の遺伝子を子に遺伝させることで、解探索性能が向上する可能性があるとも考えられる。この両面のバランスを操作し、指向性交配による解探索性能をさらに高める交叉法について検討する。

## 9.2 TNSDM-CG : 指向性交配のための交叉量操作法を用いる TNSDM

指向性交配による解探索性能をさらに向上させることを目的として、交叉において有用な実行不可能解から遺伝させる遺伝子(変数)の量を操作する方法を提案する。離散問題のためには一様交叉、連続問題には SBX を基にした方法を構築する。いずれの方法も同じコンセプトに基づいて設計される。また、これら交叉量操作を行う TNSDM を TNSDM-CG とする。図 9.3 に TNSDM-CG のブロック図を示す。基礎となる TNSDM に対して、ブロック図中の緑色で示した処理を改善する。また、TNSDM-CG の疑似コードを Algorithm 5 に示す。また、表 9.1 にブロック図と疑似コードで用いる記号を示す。Algorithm 3 と表 7.1 では、基礎となる TNSDM の記述を黒色で、TNSDM-A のための記述を緑色で示す。

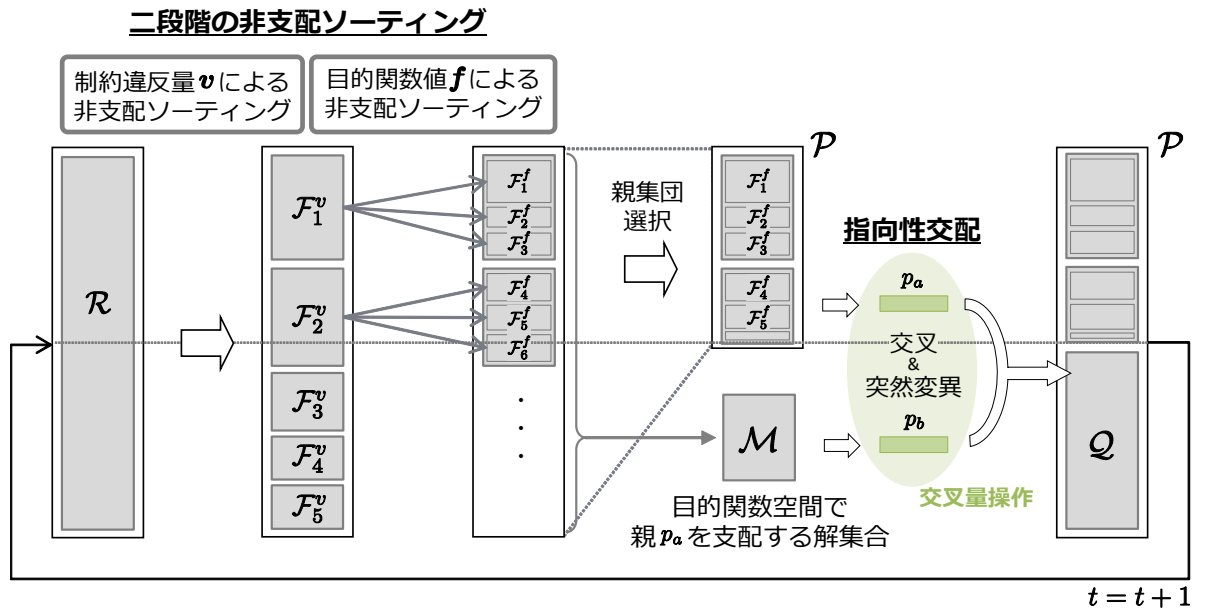


図 9.3: TNSDM-CG のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団)

### 9.2.1 離散問題のための交叉量操作

従来の一様交叉法は、二つの親の各遺伝子座の遺伝子を確率  $\gamma = 0.5$  で入れ替える。つまり、子はほぼ半分の遺伝子を第一の親  $p_a$  から、残りを第二の親  $p_b$  からコピーされる。指向性交配においては、第一の親は実行可能解、第二の親の多くは実行不可能解である。実行不可能解の親から引き継ぐ遺伝子量を変化させるために、交叉遺伝子量の操作法 (Controlling the number of Crossed Genes : CCG) [50] の概念を取り入れる。

提案法の概念図を図 9.4 に示す。第一の親  $p_a$  と第二の親  $p_b$  の遺伝子 (変数) をそれぞれ  $\mathbf{x}^{p_a} = (x_1^{p_a}, x_2^{p_a}, \dots, x_n^{p_a})$ ,  $\mathbf{x}^{p_b} = (x_1^{p_b}, x_2^{p_b}, \dots, x_n^{p_b})$  と表す。それらの子  $\mathbf{x}^o = (x_1^o, x_2^o, \dots, x_n^o)$  は以下の式によって生成される。

$$x_i^o = \begin{cases} x_i^{p_b}, & \text{if } \text{rand}[0, 1] < \gamma \\ x_i^{p_a}, & \text{otherwise} \end{cases} \quad (i = 1, 2, \dots, n), \quad (9.1)$$

$\text{rand}$  は  $[0, 1]$  の範囲のランダムな実数値である。  $\gamma$  は  $[0, 1]$  の範囲で決定するユーザ指定のパラメータであり、これによって交叉遺伝子量を操作する。  $\gamma = 0.5$  のとき、従来の一様交叉法と同じである。  $\gamma$  を  $0.5$  よりも小さくするにしたがって実行不可能解の第二の親からコピーする遺伝子の量が減少する。反対に、  $\gamma$  を  $0.5$  よりも大きくするにしたがって実行不可能解の第二の親からコピーする遺伝子の量が増加する。

この方法は、CCG 交叉法 [50] と類似している。CCG 交叉法は 3 目的以上を扱う制約のない多数目的最適化問題において、交叉する遺伝子量を減らして親の解からの変異量を

---

**Algorithm 5** TNSDM-CG

---

```
1: Initialize ( $\mathcal{R}$ )
2: for  $t = 1$  to  $T$  do
3:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
4:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sort } (\mathcal{R}, \succ_v)$ 
5:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
6:      $j = 1 + |\mathcal{F}^f|$ 
7:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sort } (\mathcal{F}_i^v, \succ_f)$ 
8:   end for
9:   Crowding distance ( $\mathcal{F}^f$ )
10:   $\mathcal{P} = \text{Truncation } (\mathcal{R}, |\mathcal{R}|/2)$ 
11:  for  $j = 1$  to  $|\mathcal{Q}|$  do
12:     $p_a = \text{Tournament selection } (\mathcal{P})$ 
13:     $\mathcal{M} = \{\mathbf{x} \in \mathcal{R} \mid \mathbf{x} \succ_f p_a\}$ 
14:    if  $p_a$  is feasible and  $|\mathcal{M}| \geq 2$  then
15:       $p_b = \text{Tournament selection } (\mathcal{M}) : \textit{Directed Mating}$ 
16:    else
17:       $p_b = \text{Tournament selection } (\mathcal{P}) : \textit{Conventional Mating}$ 
18:    end if
19:     $\mathbf{o} = \text{Crossover with exchange probability } \gamma (p_a, p_b, \gamma)$ 
20:     $Q_j = \text{Mutation } (\mathbf{o})$ 
21:  end for
22:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
23: end for
24:  $\text{POS} = \{\mathbf{x} \in \mathcal{R}^{\text{feasible}} \mid \neg \exists \mathbf{y} \in \mathcal{R}^{\text{feasible}} : \mathbf{y} \succ_f \mathbf{x}\}$ 
```

---

抑えるために提案された。これは、多数目的最適化問題において、解集団の解空間における多様性が高まるため、多くの遺伝子を組み替えるとランダムサーチ化してしまうことを避けるためである。したがって、制約なし多数目的最適化問題において、交叉する親は二つとも実行可能解であり、小さな  $\gamma$  を用いて交叉遺伝子量を抑えた場合に解探索性能が向上することがわかっている。一方で、本研究では制約付き多目的最適化問題の TNSDM での解探索において、交叉で実行不可能解から受け継ぐ遺伝子の量を操作する。

表 9.1: 擬似コード内で使用される記号 (TNSDM-CG)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$o$	交叉により生成した子
$\gamma$	交叉量を操作するパラメータ
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

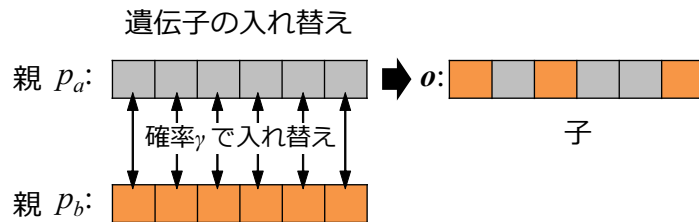


図 9.4: 一様交叉を基にした交叉量操作

## 9.2.2 連続問題のための交叉量操作

従来の SBX 交叉法では, まず, 第一の親  $p_a$  と第二の親  $p_b$  の遺伝子 (変数) を変化させ,  $p_a'$  と  $p_b'$  を生成する. 各遺伝子の変異量は二つの親の各遺伝子座の変数値の差と分布パラメータ  $\eta_c (= [0, \infty])$  によって決定される確率分布関数に従う. 親同士の変数値が近いほど, また, パラメータ  $\eta_c$  の値が小さいほど変異量は小さくなりやすい. 次に,  $p_a'$  と  $p_b'$  の各変数を  $\gamma = 0.5$  の確率で入れ替え, 子個体とする. この遺伝子の入れ替えは解

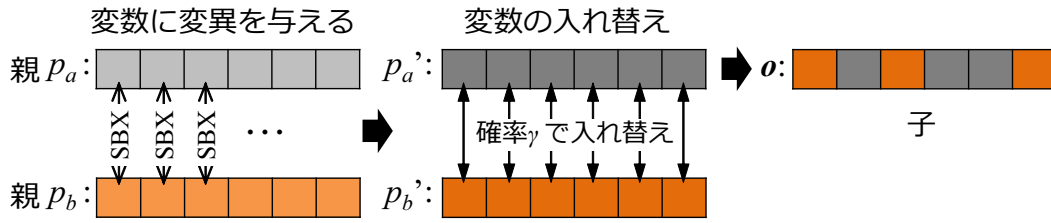


図 9.5: SBX を基にした交叉量操作

探索性能の向上に貢献することが報告されている [49, 51]. この SBX 交叉法を基にして、指向性交配において実行不可能解の親から引き継ぐ遺伝子量を変化させる概念を提案の離散問題の交叉法と同様に導入する.

提案法 の 概念 図 を 図 9.5 に 示 す . ま ず , 指 向 性 交 配 に お け る 実 行 可 能 解 の 親  $p_a$  と 実 行 不 可 能 解 の 親  $p_b$  を SBX 交叉法と同じ式を用いて遺伝子に変異を与え,  $p_a'$  と  $p_b'$  とする. その後, それらの変数ベクトルを  $\mathbf{x}^{p_a'} = (x_1^{p_a'}, x_2^{p_a'}, \dots, x_n^{p_a'})$ ,  $\mathbf{x}^{p_b'} = (x_1^{p_b'}, x_2^{p_b'}, \dots, x_n^{p_b'})$  と表し, 子  $\mathbf{x}^o = (x_1^o, x_2^o, \dots, x_n^o)$  を以下の式によって生成する.

$$x_i^o = \begin{cases} x_i^{p_b'}, & \text{if } \text{rand}[0, 1] < \gamma \quad (i = 1, 2, \dots, n), \\ x_i^{p_a'}, & \text{otherwise} \end{cases} \quad (9.2)$$

$\text{rand}$  は  $[0, 1]$  の範囲のランダムな実数値である.  $\gamma$  は  $[0, 1]$  の範囲で決定するユーザ指定のパラメータであり, これによって交叉遺伝子量を操作する.  $\gamma = 0.5$  のとき, 従来の SBX 交叉法と同じである.  $\gamma$  を 0.5 よりも小さくするにしたがって実行不可能解の第二の親を基にした遺伝子をコピーする量が減少する. 反対に,  $\gamma$  を 0.5 よりも大きくするにしたがって実行不可能解の第二の親を基にした遺伝子をコピーする量が増加する.

### 9.3 実験内容

TNSDM-CG の解探索性能を指向性交配において実行可能解と実行不可能解の親の交叉量  $\gamma$  を変化させて検証する. なお, 指向性交配ができない場合 (第一の親を支配する第二親候補集合  $\mathcal{M}$  に解が 2 つ以上存在しない場合) には, 第二の親を親集団  $\mathcal{P}$  から選択する従来の交配法を用いるが, このときの交叉は  $\gamma = 0.5$  の従来の一様交叉あるいは SBX 交叉法を用いる. TNSDM-CG の性能は, 指向性交配において従来の一様交叉あるいは SBX 交叉法を用いる TNSDM, さらに, すべての交配で従来の交配法を行う TNSDM(交叉法も従来の一様交叉あるいは SBX 交叉法) と比較する. またテスト問題として, 離散問題は  $mk$ -KP を使い, 目的数と実行可能率を変化させて性能を検証する. 連続問題としては TNK と mCDTLZ を使い, mCDTLZ では目的数 (=制約数) を変化させて性能を検証する.

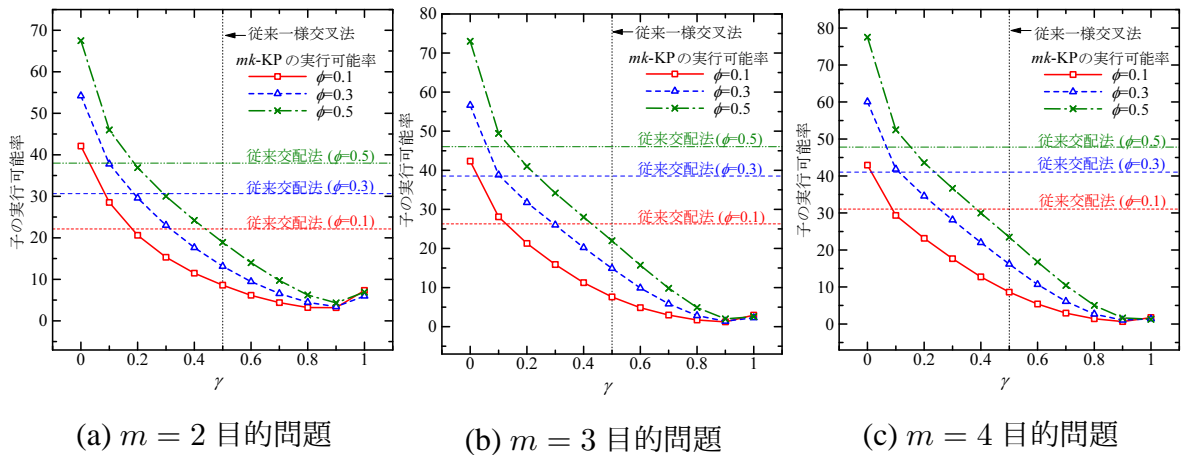


図 9.6:  $mk$ -KP における生成された子の実行可能率

また、実行不可能解の親の遺伝子量の影響を分析するために、全世代を通じて生成された子の実行可能率について調べる。

### 9.3.1 パラメータ

$mk$ -KP の問題パラメータは、アイテム数  $n = 500$ 、目的数  $m = \{2, 3, 4\}$ 、ナップサック (制約) 数  $k = 6$ 、実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  に設定した。 $mk$ -KP におけるアルゴリズムの遺伝的パラメータは、一様交叉法 (交叉率  $P_c = 1.0$ )、ビット反転突然変異法 (突然変異率  $P_m = 1/n$ )、世代数  $T = 10,000$  を用いた。ただし、指向性交配においては、交叉法は CCG を用いた Figure/TNSDM-CG を用いる。また、集団サイズ  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) とした。実験の結果は 50 試行の平均値を用いて示す。

mCDTLZ の問題パラメータは、目的数 (=制約数)  $m = \{2, 3, 4\}$ 、変数の数  $n = 12$  とした。TNK と mCDTLZ 問題におけるアルゴリズムの遺伝的パラメータは、SBX 交叉法 (交叉率  $P_c = 0.8$ 、分布パラメータ  $\eta_c = 15$ )、Polynomial mutation (突然変異率  $P_m = 0.2$ 、分布パラメータ  $\eta_m = 20$ )、世代数は TNK では  $T = 1,000$ 、mCDTLZ では  $T = 5,000$  を用いる。集団サイズは  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ) とした。実験の結果は 100 試行の平均値を用いて示す。

### 9.3.2 評価尺度

得られたパレート最適解集合の評価は、総合評価尺度である Hypervolume ( $HV$ ) を用いる。 $HV$  の算出に用いる参照点  $\mathbf{r}$  は、 $mk$ -KP では  $\mathbf{r} = (0, 0, \dots, 0)$ 、TNK では  $\mathbf{r} = (1.2, 1.2)$ 、mCDTLZ では  $\mathbf{r} = (1.1, 1.1, \dots, 1.1)$  を用いる。

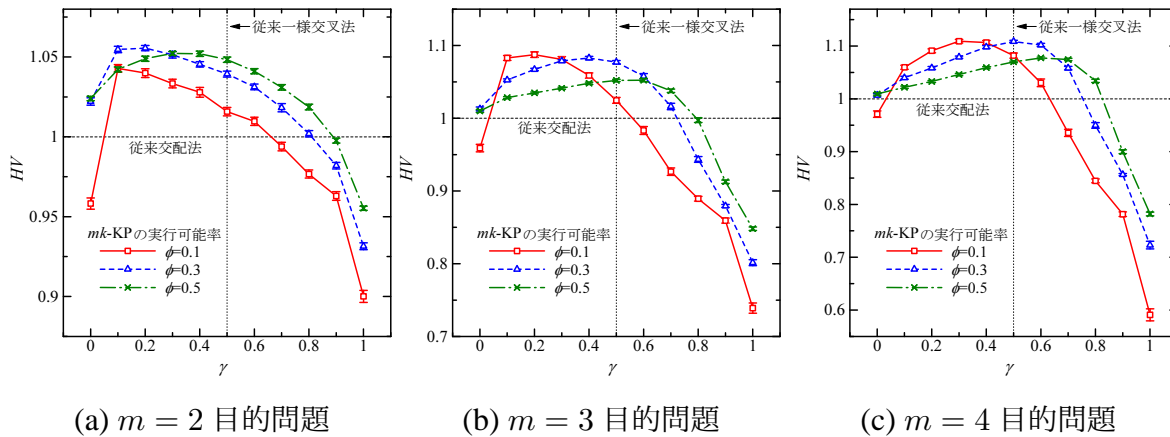


図 9.7:  $mk$ -KP における最終世代の  $HV$

## 9.4 実験結果と考察

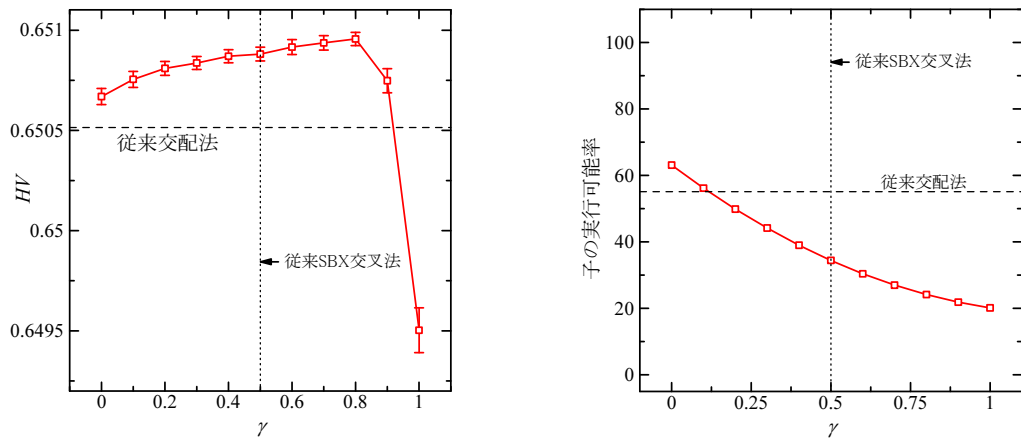
### 9.4.1 $mk$ -KP における生成された子の実行可能率

まず、離散問題の  $mk$ -KP において、TNSDM-CG の  $\gamma$  を変化させた場合の、指向性交配で生成された子の実行可能率を調べる。指向性交配で生成した子の実行可能率を図 9.6 に示した。また、すべての親を親集団  $\mathcal{P}$  から選択する従来の交配法（一様交叉  $\gamma = 0.5$ ）を用いた場合の結果も水平線でプロットした。これらの結果から、まず、従来の一様交叉（ $\gamma = 0.5$ ）を指向性交配と組み合わせて用いる場合、すべての問題において、生成される子の実行可能率は、従来の交配法より低くなるのがわかる。しかし、TNSDM-CG における  $\gamma$  の減少に伴い、実行不可能な第二の親から複写される遺伝子数が減少することにより、生成される子の実行可能率が改善することがわかる。また、 $\gamma$  の増加に伴い実行不可能解の親から複写される遺伝子数が増加することで、生成される子の実行可能率が低下することがわかる。

### 9.4.2 $mk$ -KP における解探索性能

次に、離散問題の  $mk$ -KP において、TNSDM-CG の  $\gamma$  を変化させた場合の解探索性能 ( $HV$ ) の結果を図 9.7 に示す。それぞれの図において、すべての交配で従来の交配法（一様交叉  $\gamma = 0.5$ ）を用いた場合の TNSDM の結果を 1.0 として正規化し、水平線でプロットした。また、TNSDM-CG の結果のエラーバーは 95% 信頼区間を示している。また、TNSDM-CG において  $\gamma = 0.5$  を用いる結果は指向性交配を用いる通常の TNSDM に等しい。





(a) 解探索性能 ( $HV$ )

(b) 生成された子の実行可能率

図 9.8: TNK における最終世代の  $HV$  と子の実行可能率

この結果から、まず、それぞれの  $mk$ -KP において、TNSDM-CG は解探索性能  $HV$  を最大化する最適な  $\gamma^*$  が存在することがわかる。  $\gamma$  が  $\gamma^*$  より小さくなると、図 9.6 に示したように生成される子の実行可能率は改善するが、  $HV$  は低下することがわかる。これは、生成される子が実行可能解になりやすかったとしても、第二の親から複製される解探索を促進する遺伝子 (変数) 情報が減少するためと考えられる。また、  $\gamma$  が  $\gamma^*$  より大きくなる時も、  $HV$  は低下する。これは、  $\gamma$  を大きくすることによって、第二の実行不可能解の親から複製される遺伝子 (変数) の数は増えることによって、図 9.6 に示すように生成された子が実行不可能解になりやすいためと考えられる。

また、  $HV$  を最大化する  $\gamma^*$  は、それぞれの  $mk$ -KP によって変わることがわかる。実行可能率が高い  $\phi = 0.5$  の  $m = \{3, 4\}$  目的問題では、最も高い  $HV$  が  $\gamma^* > 0.5$  で達成されることがわかる。このとき、実行不可能な第二の親の遺伝子 (変数) の半分以上が子に複製される。このケースでは、図 9.6 に示すように生成される子の実行可能率が比較的高いため、有用な実行不可能解からより多くの遺伝子を複製したときに、高い解探索性能が達成されるといえる。

### 9.4.3 TNK 問題における結果

連続問題の TNK 問題における結果について議論する。TNSDM-CG の  $\gamma$  を変化させた場合の解探索性能  $HV$  の結果を図 9.8 (a) に示す。ここで、エラーバーは 95% 信頼区間を示している。前節で議論した離散問題とは異なり、ここでは SBX に基づく方法で変数交換率  $\gamma$  を操作することに注意されたい。  $\gamma = 0.5$  を用いる従来の交配法によって達成された  $HV$  を水平線で示した。さらに、従来の遺伝子交換率  $\gamma = 0.5$  を垂直線で示す。また、  $\gamma = 0.5$  を用いる TNSDM-CG は、TNSDM と等しいことに注意されたい。  $\gamma$  を 0.5 より

減少させると、有用な実行不可能解の親から複写される変数情報が減少する。逆に、 $\gamma$  を 0.5 より増加させると、有用な実行不可能解の親から複写される変数情報が増加する。

まず、水平線で示される従来の交配法と  $\gamma = 0.5$  を用いる指向性交配を比較すると、後者が高い  $HV$  を示すことがわかる。これは、 $\gamma$  によって変数交換率を操作しない従来の指向性交配による解探索効果である。次に、TNSDM-CG における  $\gamma$  を 0.5 から増加させることによって  $HV$  が改善し、最も高い  $HV$  は  $\gamma^* = 0.8$  で達成されることがわかる。この結果から、TNSDM-CG において、有用な実行不可能解の親から複写される変数情報を増加させた場合に、指向性交配による解探索効果がさらに高まることがわかった。次に、 $\gamma = \{0, 1\}$  の 2 ケースについて述べる。 $\gamma = 0$  の場合、すべての変数は交換されず、SBX によって変異した  $p_a'$  が、そのまま子になる。 $\gamma = 1$  の場合、すべての変数が交換され、SBX によって変異した  $p_b'$  が、そのまま子になる。図 9.8 (a) の結果から、 $\gamma = 0.5$  を用いる TNSDM-CG は、 $\gamma = \{0, 1\}$  を用いる二つの TNSDM-CG より高い  $HV$  を示すことがわかり、SBX に基づく交叉法において、各変数を交換することが解探索性能の改善に寄与していることが確かめられた。また、図 9.8 (a) の結果から、変数交換率  $\gamma$  を操作することによって、提案する TNSDM-CG の解探索性能が改善することが確かめられた。

次に、指向性交配によって得られた子の実行可能率を図 9.8 (b) に示す。この結果から、 $\gamma = 0.5$  を用いる従来の指向性交配による子の実行可能率は、従来の交配法より低いことがわかる。これは、 $mk$ -KP における結果と同様で、従来の指向性交配によって生成される子は実行不可能解になりやすいことを示している。 $\gamma$  を 0.5 より増加させると、指向性交配によって生成される子の実行可能率はさらに減少するが、図 9.8 (a) に示すように  $HV$  は改善することがわかる。すなわち、大きな  $\gamma$  によって生成された子は実行不可能解になりやすいが、良い目的関数値を示す有用な実行不可能解の親の変数情報を繰り返し指向性交配によって遺伝させることによって、解探索性能が高まったといえる。

#### 9.4.4 mCDTLZ 問題における結果

次に、連続問題の mCDTLZ 問題における結果について議論する。 $m = \{2, 3, 4\}$  目的の mCDTLZ において、TNSDM-CG の  $\gamma$  を変化させた場合の解探索性能  $HV$  の結果を図 9.9 (a) に示す。また、エラーバーは 95% 信頼区間を示している。なお、すべての結果は、 $\gamma = 0.5$  を用いる従来の交配法による  $HV$  を 1.0 として正規化してプロットしている。

TNK 問題の結果と同様に、 $\gamma$  を 0.5 から増加させて、有用な実行不可能解の親から子に複写される変数情報を増加させた場合に、 $HV$  が改善することがわかる。また、 $HV$  の改善効果は、目的数の増加に伴って顕著になることがわかる。ただし、信頼区間も広がってしまうことがわかる。6.4.4 節の図 6.10 に示されるように、mCDTLZ において目的数が増加すると、TNSDM では指向性交配を行う回数が大きく減少してしまうことがわかっている。数少ない指向性交配においてのみ交叉量操作を行うため、効果の影響が少なく、

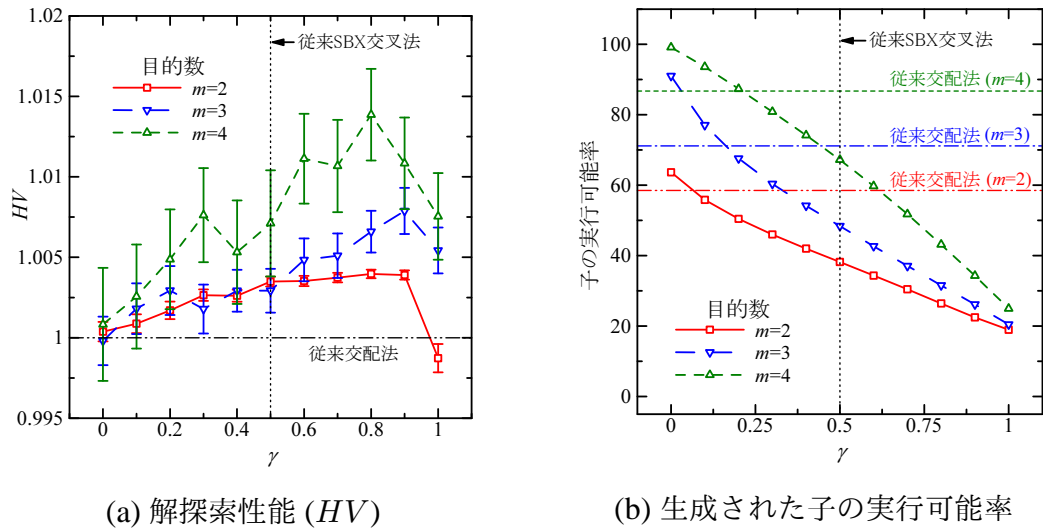


図 9.9: mCDTLZ における最終世代の  $HV$  と子の実行可能率

安定しないことが理由と考えられる。しかし、 $HV$  を最大化する最適な  $\gamma^*$  の結果は、従来の交叉法と等しい  $\gamma = 0.5$  の結果と比べて有意差があるといえる。

指向性交配によって生成された子の実行可能率を図 9.9 (b) に示す。この結果から、まず、目的数の増加に伴って生成される子の実行可能率が高まる傾向が見て取れる。これは、目的数の増加に伴い、目的空間の実行可能領域と不可能領域の境界にあるパレートフロントへ解集団が近づくことが徐々に困難になることが原因と考えられる。また、 $\gamma$  が変化すると、TNK 問題で得られた傾向と同様の結果が見られることがわかった。

## 9.5 まとめ

指向性交配による解探索効果をさらに高めるため、本章では、子を生成するときの交叉法に注目し、有用な実行不可能解の親から子に複製される遺伝子(変数)量を操作する方法を提案した。離散問題においては、一様交叉を基礎とし、二つの親の各遺伝子を交換する確率を  $\gamma$  で操作する方法を提案した。連続問題においては、SBX 交叉を基礎とし、二つの親の各遺伝子に SBX 変異を与えたあとに、遺伝子を交換する確率を  $\gamma$  によって操作する方法を提案した。いずれの方法も、 $\gamma = 0.5$  ときに従来の交叉法と同じ効果が得られる。 $\gamma$  を 0.5 より小さくすると、有用な実行不可能解の親から子に複製される遺伝子(変数)量が減少する。このとき、生成された子は実行可能解になりやすいが、有用な実行不可能解の親から複製される目的関数値を高めるための遺伝子情報は得られにくくなる。逆に、 $\gamma$  を 0.5 より大きくすると、有用な実行不可能解の親から子に複製される遺伝子(変数)量が増加する。このとき、生成された子は実行不可能解になりやすいが、有用な実行

不可能解の親から複製される目的関数値を高めるための遺伝子情報が増加する。この方法を TNSDM に組み込んだアルゴリズムを TNSDM-CG とし、離散問題の *mk-KP* と連続問題の *TNK*, *mCDTLZ* を用いて解探索性能を検証した結果、いずれの問題においても  $\gamma$  を適切な値に設定することにより、指向性交配による解探索効果がさらに高まることが確かめられた。ただし、問題ごとに解探索性能を最大化する  $\gamma$  は異なることがわかった。 $\gamma$  を小さくすると、子は実行可能解になりやすいが有用な実行不可能解の遺伝子情報が得られにくくなることと、 $\gamma$  を大きくすると、子は有用な実行不可能解の遺伝子情報を得やすいが実行不可能解になりやすいことのバランスによるものと考えられる。

## 第 10 章

# 有用な実行不可能解の選出領域制御法とアーカイブ法，交叉量操作法の組み合わせ

### 10.1 TNSDM-CSACG：選出領域制御法とアーカイブ法，交叉量操作法を用いる TNSDM

本章では，TNSDM に有用な実行不可能解の選出領域制御法とアーカイブ法，さらに交叉量操作法を組み合わせた TNSDM-CSACG を提案する．図 10.1 に TNSDM-CSACG のブロック図を示す．また，TNSDM-CSACG の疑似コードを **Algorithm 6** に示す．ブロック図と疑似コード中では，基礎となる TNSDM に対して，有用な実行不可能解の選出領域制御の処理を赤色で，アーカイブに関する処理を青色，交叉量操作に関する処理を緑色で示す．また，表 10.1 にブロック図と疑似コードで用いる記号を示す．TNSDM-CSACG のアルゴリズムは，8 章で提案された TNSDM-CSA において，指向性交配での子個体生成時に交叉量操作する方法といえる．

TNSDM-CSACG は，3 つのパラメータを必要とする．一つ目は，有用な実行不可能解の選出領域を制御するためのパラメータ  $S$  である．二つ目は，アーカイブサイズのパラメータ  $\alpha$ ，三つ目は交叉量を操作するためのパラメータ  $\gamma$  である．

このように，TNSDM を発展させる選出領域制御法，アーカイブ法，交叉量操作法は，それぞれの処理が競合することなく，親和的に一つのアルゴリズムの中に組み込める．

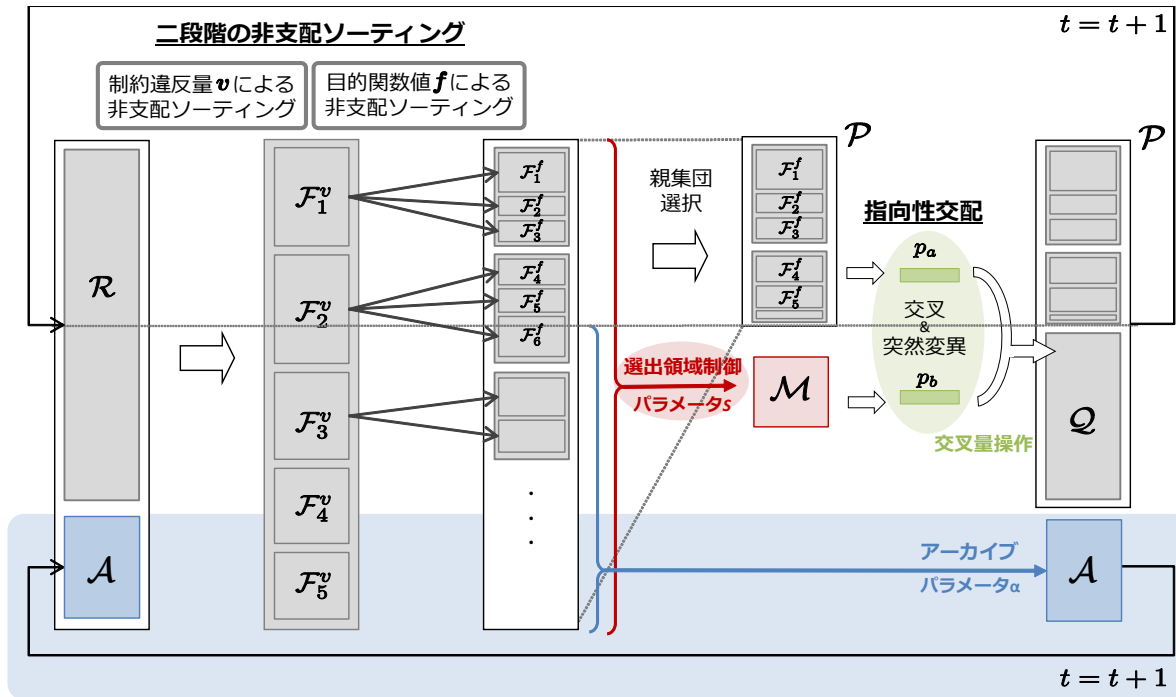


図 10.1: TNSDM-CSACG のブロック図 ( $\mathcal{R}$ : 全解集団,  $\mathcal{P}$ : 親集団,  $\mathcal{Q}$ : 子集団,  $\mathcal{A}$ : アーカイブ集団))

## 10.2 実験内容

$mk$ -KP を用いて, 実行不可能解の選出領域制御とアーカイブ, 交叉量操作を用いる TNSDM (TNSDM-CSACG) において, TNSDM-CSA において最も  $HV$  値の高まる最適な選出領域制御のパラメータ  $S$  とアーカイブサイズのパラメータ  $\alpha$  を用いたときに, 交叉量を操作するパラメータ  $\gamma$  を変化させた場合の解探索性能について検証する. 結果の比較には, TNSDM, TNSDM-CS, TNSDM-A, TNSDM-CSA, TNSDM-CG と従来法の CNSGA-II を用いる. これにより, TNSDM を発展させる選出領域制御法, アーカイブ法, 交叉量操作法の解探索性能に対する貢献度を確認するとともに, これらが TNSDM の中で協調して解探索性能の改善に貢献することを確認する.

### 10.2.1 パラメータ

$mk$ -KP のパラメータは, アイテム数  $n = 500$ , 目的数  $m = \{2, 4\}$ , 制約数  $k = 6$ , 実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  とした. 遺伝的パラメータとして, 解集団サイズは,  $|\mathcal{R}| = 200$  ( $|\mathcal{P}| = |\mathcal{Q}| = 100$ ), 世代数  $T = 10,000$  世代, ビット反転突然変異法 (変異率  $P_m = 1/n$ ) を用いる. 交叉法は, TNSDM-CSACG と TNSDM-CG において従来 of 交配を行う場合

---

**Algorithm 6** TNSDM-CSACG

---

```
1: Randomly generate initial solutions ( $\mathcal{R}$ )
2:  $\mathcal{A} = \emptyset$ 
3: for  $t = 1$  to  $T$  do
4:    $\mathcal{F}^v = \mathcal{F}^f = \emptyset$ 
5:    $\mathcal{F}^v (= \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots\}) = \text{Non-dominated sort } (\mathcal{R} \cup \mathcal{A}, \succ_v)$ 
6:   for  $i = 1$  to  $|\mathcal{F}^v|$  do
7:      $j = 1 + |\mathcal{F}^f|$ 
8:      $\mathcal{F}^f = \mathcal{F}^f \cup \{\mathcal{F}_j^f, \mathcal{F}_{j+1}^f, \dots\} = \text{Non-dominated sort } (\mathcal{F}_i^v, \succ_f)$ 
9:   end for
10:  Crowding distance ( $\mathcal{F}^f$ )
11:   $\mathcal{P} = \text{Truncation } (\mathcal{R} \cup \mathcal{A}, |\mathcal{R}|/2)$ 
12:  Calculate  $f'$  from  $f$  ( $\mathcal{R} \cup \mathcal{A}, S$ )
13:  for  $j = 1$  to  $|\mathcal{Q}|$  do
14:     $p_a = \text{Tournament selection } (\mathcal{P})$ 
15:     $\mathcal{M} = \{x \in \mathcal{R} \cup \mathcal{A} \mid x \succ_{f'} p_a\}$ 
16:    Mark best  $\alpha$  solutions selected in  $\mathcal{M}$  ( $\{x \in \mathcal{R} \cup \mathcal{A} \mid x \succ_{f'} p_a\}$ )
17:    if  $p_a$  is feasible or  $|\mathcal{M}| \geq 2$  then
18:       $p_b = \text{Tournament selection } (\mathcal{M}) : \textit{Directed Mating}$ 
19:    else
20:       $p_b = \text{Tournament selection } (\mathcal{P}) : \textit{Conventional Mating}$ 
21:    end if
22:     $o = \text{Crossover with exchange probability } \gamma (p_a, p_b, \gamma)$ 
23:     $Q_j = \text{Mutation } (o)$ 
24:  end for
25:   $\mathcal{A} = \{x \in \mathcal{R} \cup \mathcal{A} \mid x \text{ is a marked solution} \wedge x \notin \mathcal{P}\}$ 
26:   $\mathcal{R} = \mathcal{P} \cup \mathcal{Q}$ 
27: end for
28:  $\mathcal{POS} = \{x \in \mathcal{R}^{feasible} \mid \neg \exists y \in \mathcal{R}^{feasible} : y \succ_f x\}$ 
```

---

と、その他アルゴリズムにおいては  $\gamma = 0.5$  の従来の一様交叉 (交叉率  $P_c = 1.0$ ) を用いる。世代数  $T = 10,000$  世代、ビット反転突然変異法 (変異率  $P_m = 1/n$ ) を用いる。交叉法は、TNSDM-CSACG と TNSDM-CG において従来 of 交配を行う場合と、その他アルゴリズムにおいては  $\gamma = 0.5$  の従来の一様交叉 (交叉率  $P_c = 1.0$ ) を用いる。TNSDM-CSACG と TNSDM-A の計算コストは、アーカイブのパラメータ  $\alpha$  によって増

表 10.1: 擬似コード内で使用される記号 (TNSDM-CSACG)

$\mathcal{P}$	親集団. 親集団サイズは $ \mathcal{P}  (= \mathcal{R} /2)$ .
$\mathcal{Q}$	子集団. 子集団サイズは $ \mathcal{Q}  (= \mathcal{R} /2)$ . 各個体は $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{ \mathcal{Q} }\}$ .
$\mathcal{R}$	全解集団. $\mathcal{P} \cup \mathcal{Q}$ .
$\mathcal{A}$	有用な実行不可能解のアーカイブ集団
$\alpha$	アーカイブサイズパラメータ
$\mathcal{F}^v$	全解集団 $\mathcal{R}$ を制約違反量 $v$ で分類したフロント群で, $\mathcal{F}^v = \{\mathcal{F}_1^v, \mathcal{F}_2^v, \dots, \mathcal{F}_{ \mathcal{F}^v }^v\}$ . $ \mathcal{F}^v $ はフロント数.
$\mathcal{F}^f$	$\mathcal{F}^v$ をさらに目的関数値 $f$ で分類したフロント群で, $\mathcal{F}^f = \{\mathcal{F}_1^f, \mathcal{F}_2^f, \dots, \mathcal{F}_{ \mathcal{F}^f }^f\}$ . $ \mathcal{F}^f $ はフロント数.
$S$	第二の親の候補解集合 $\mathcal{M}$ の選出領域を制御するパラメータ
$p_a, p_b$	子を生成するための親のペア
$\mathcal{M}$	親 $p_a$ を目的関数空間で支配する解集合
$o$	交叉により生成した子
$\gamma$	交叉量を操作するパラメータ
$T$	探索終了条件である世代数
$\mathcal{R}^{feasible}$	全解集団 $\mathcal{R}$ 中の実行可能解集合. $\mathcal{R}^{feasible} \subseteq \mathcal{R}$ .
$POS$	MOEA の結果として出力するパレート最適解集合

加するが, 本章ではすべてのアルゴリズムを同じ世代数  $T$ (同じ解の評価回数) のもとで比較することとする. 結果は, 各アルゴリズムの 50 回の試行の平均値を用いて比較する.

## 10.2.2 評価尺度

獲得した解集合を総合評価する Hypervolume( $HV$ ) によって, TNSDM-CSACG の性能を検証する.  $HV$  を算出する参照点  $r$  は, 目的空間の原点  $r = (0, 0, \dots, 0)$  に設定した.

## 10.3 実験結果と考察

各アルゴリズムによって得られた最終世代の  $HV$  の結果を図 10.2 と図 10.3 に示す. すべての結果は CNSGA-II によって得られた値を 1.0 として正規化している. ここで, TNSDM-CS, TNSDM-A, TNSDM-CSA(TNSDM-CSACG) において手法に必要なパラメータは最も  $HV$  が高まる最適な設定を用いており, その値は図中に示している. それ



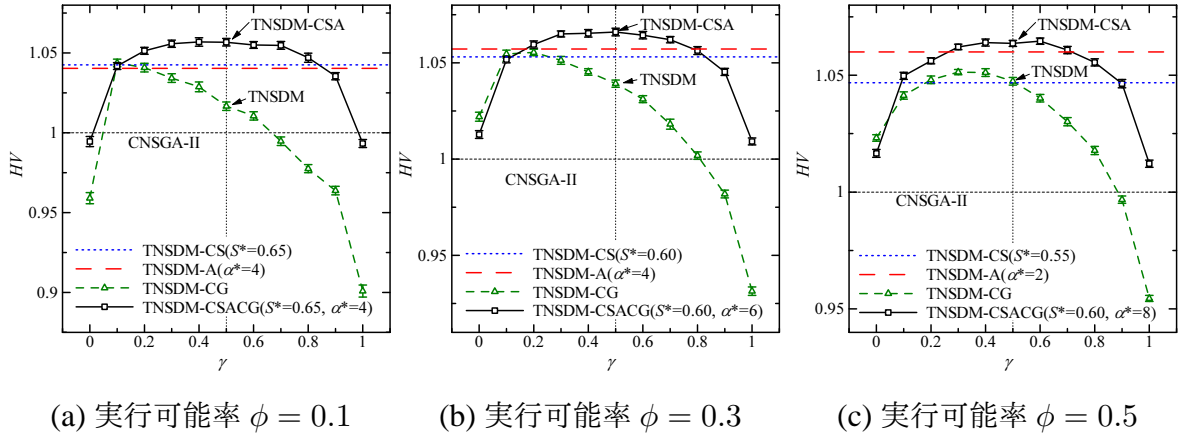


図 10.2: 最終世代の  $HV(m = 2)$  目的問題

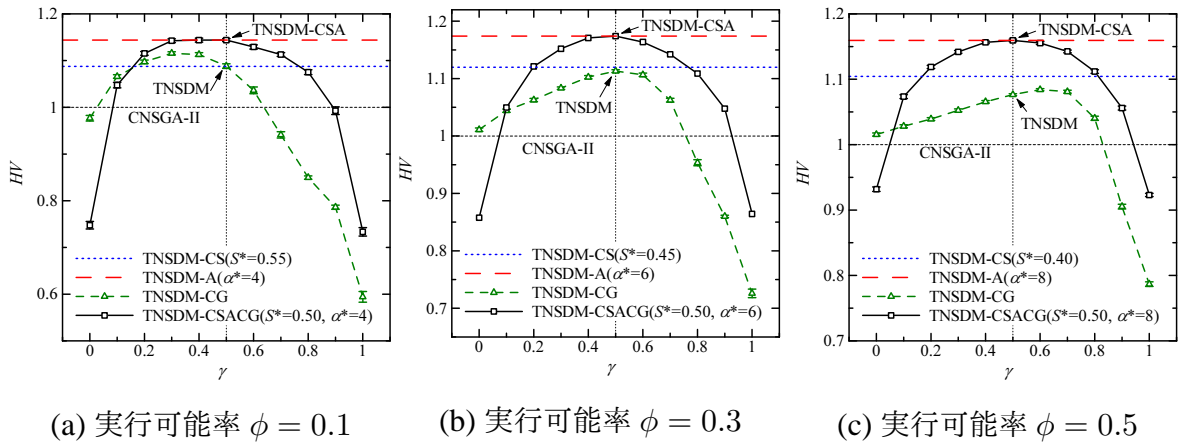


図 10.3:  $mk$ -KP における最終世代の  $HV(m = 4)$  目的問題

それぞれの図において、TNSDM-CG と TNSDM-CSACG において交叉量操作のパラメータ  $\gamma$  を変化させた場合の結果をプロットしている。TNSDM-CG において  $\gamma = 0.5$  の結果は TNSDM に等しく、TNSDM-CSACG において  $\gamma = 0.5$  の結果は TNSDM-CSA に等しい。これらの結果の 95% 信頼区間もエラーバーで示している。また、交叉量操作を含まない TNSDM-CS, TNSDM-A, CNSGA-II においては  $\gamma = 0.5$  の通常の一様交叉を用いた結果であり、水平線で示している。

まず、これまでの章で示された通り、基本アルゴリズムである TNSDM は従来の CNSGA-II よりも、すべての問題で高い  $HV$  を示すことがわかる。さらに、TNSDM に選出領域制御を加えた TNSDM-CS, アーカイブを加えた TNSDM-A, 交叉量操作を加える TNSDM-CG はそれぞれのパラメータを最適に設定することによって TNSDM よりも高い  $HV$  を示し、解探索性能を高める効果があることがわかる。また、選出領域制御を加えた TNSDM-CS, アーカイブを加えた TNSDM-A を組み合わせた TNSDM-CSA は、

それぞれ組み合わせない場合と比べて高い  $HV$  を示している。ただし、TNSDM-CSA において最適な選出領域制御パラメータが  $S^* = 0.5$  である問題では、TNSDM-A と同じ値となっている。次に TNSDM-CSACG について、 $HV$  を最も高める最適な  $\gamma$  は  $m = 2$  目的問題の実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の問題でそれぞれ  $\gamma^* = \{0.4, 0.5, 0.6\}$ ,  $m = 4$  目的問題の実行可能率  $\phi = \{0.1, 0.3, 0.5\}$  の問題でそれぞれ  $\gamma^* = \{0.4, 0.5, 0.5\}$  となっている。 $\gamma = 0.5$  を用いる TNSDM-CSA と比べて効果は小さいが、TNSDM-CSACG が最も高い  $HV$  を示すことがわかった。また、TNSDM-CG において最適な  $\gamma^*$  が  $0.5$  以下と、実行不可能解から複写する遺伝子量を減少した場合に  $HV$  が高まる問題において、TNSDM-CSACG によって指向性交配における有用な実行不可能解の選出領域制御とアーカイブを加えることによって、最適な  $\gamma^*$  の値が大きくなっていることがわかる。すなわち、実行不可能解から受け継ぐ遺伝子量を増加させた方が良い子個体が生成されやすく、解探索性能が高まったといえる。このことから、アーカイブ法により解集団中に保持している実行不可能解、また、選出領域制御法により交配する親に選ばれる実行不可能解の持つ遺伝子情報は、これらを用いない場合と比べてより有用な実行不可能解であるといえる。有用な実行不可能解をアーカイブ保存しない TNSDM-CG では、交配に用いる実行不可能解は前世代に子個体として生成された解のみであるが、アーカイブを用いる TNSDM-CSACG では、アーカイブ集団として実行不可能解が解探索の過程で生じた解の中で良いもので更新されながら保持されるため、実行不可能解がより制約違反量の小さく目的関数値の良い解へと進化する。また、TNSDM-CSACG での選出領域制御法では、 $m = 2$  目的問題において、パラメータ  $S^* > 0.5$  の縮小された選出領域を用いており、親として選択された実行可能解の複数の目的関数値を等しく高める方向への進化の指向性を高めるための実行不可能解が交配相手として選出される。

選出領域制御法、アーカイブ法、交叉量操作法の  $HV$  に対する改善効果の大きさは、問題によって異なる。 $m = 2$  目的の実行可能率  $\phi = \{0.1, 0.3\}$  の問題では、選出領域制御法、アーカイブ法、交叉量操作法のそれぞれによる  $HV$  の改善効果は、同程度である。それ以外の問題では、アーカイブ法が最も  $HV$  の改善に貢献しているが、選出領域制御法と交叉量操作法による  $HV$  の改善効果の大きさは問題による。このように、選出領域制御法、アーカイブ法、交叉量操作法の  $HV$  に対する改善効果の大きさは、問題によって異なることがわかる。ただし、全ての問題において、選出領域制御法、アーカイブ法、交叉量操作法のそれぞれが  $HV$  を高める効果をもたらし、それらを組み合わせた TNSDM-CSA と TNSDM-CSACG において、それぞれが互いに妨げにならず、親和的に協調し、最も高い  $HV$  を達成できたといえる。

## 10.4 まとめ

6章で提案した有用な実行不可能解の選出領域制御法と、7章で提案した有用な実行不可能解のアーカイブ法、9章で提案した交叉量操作法を組み合わせた TNSDM-CSACG を提案した。 *mk-KP* を用いた解探索性能の検証の結果、TNSDM-CSACG は、選出領域制御法のみを用いる場合とアーカイブ法のみを用いる場合、交叉量操作法のみを用いる場合より、高い解探索性能を示すことが確かめられた。

今回用いた *mk-KP* の範疇でも、選出領域制御法、アーカイブ法、交叉量操作法、それらを組み合わせた場合の *HV* の改善効果の大きさは変化した。そのため、他の問題においても、これらそれぞれの解探索性能への貢献度は変化することが予想される。TNSDM の効果を高める選出領域制御法、アーカイブ法、交叉量操作法は、それぞれの処理が競合することなく親和的に一つのアルゴリズムに取り込めること、さらにそれぞれが互いに妨げにならずに、解探索を促進できるところに利点があると考えられる。

# 第 11 章

## 結論

### 11.1 得られた知見

本研究では，制約条件を有する多目的最適化問題の進化計算による解法において，これまでの従来法では，解集団から排除されてきた制約条件を満たさない実行不可能解を解探索に活用して解探索性能を高める方法について探求した．具体的には，実行不可能解の中には，制約条件は満たさないものの，実行可能解より良い目的関数値を有するものがあることに着眼し，これを解探索の新しい手掛かりとするために親として選択して子生成に活用する方法を提案した．離散問題として組み合わせ最適化の代表的なモデルである  $m$  目的  $k$  ナップザック問題と，近年盛んな工学設計の最適化で採用される解を実数値で表現する連続問題として，SRN, TNK, OSY, mCDTLZ, さらに実問題の 3 段式ハイブリッドロケットの概念設計最適化問題を用い，提案法の効果を従来法と比較しながら検証した．その結果，本研究で提案するいずれのアルゴリズムも，従来法より高い解探索性能を示すことを明らかにした．その詳細について，以下に述べる．

#### TNSDM

本研究の基礎アルゴリズムとなる TNSDM を提案した．TNSDM は，制約条件を満たさない実行不可能解を目的関数値の良好な実行可能解へ進化させるための解のランキングを構築する二段階の非支配ソートと，実行可能解とそれより目的関数値が良い実行不可能解を親として選択して子を生成する指向性交配を用いる．テスト問題を用いた解探索性能の検証実験の結果，提案する TNSDM は，従来法の CNSGA-II と RTS-O, RTS-C より高い解探索性能を示すことがわかった．実行不可能解を親にすることを避ける従来法との比較から，良好な目的関数値を有する実行不可能解を活用して子生成することの有効性が明らかになった．

### TNSDM-CS : 有用な実行不可能解の選出領域制御法

TNSDM における指向性交配の解探索効果をさらに高めるため、目的空間における有用な実行不可能解の選出領域をパラメータ  $S$  によって制御する方法を提案した。この方法では、有用な実行不可能解の選出領域を縮小することによって、指向性交配における解探索の指向性を強化できる。逆に、選出領域を拡大することによって、選出領域に有用な実行不可能解が含まれやすくなり、指向性交配の実行回数を増加させる効果がある。この方法を TNSDM に組み込んだアルゴリズムを TNSDM-CS とし、テスト問題を用いて解探索性能を検証した結果、有用な実行不可能解の選出領域を制御することによって、指向性交配による解探索性能の改善効果が高まることが明らかになった。解探索中に生成される有用な実行不可能解の数が多い問題では、選出領域を縮小して解探索の指向性を高めた場合に解探索性能が改善されることがわかった。逆に、解探索中に生成される有用な実行不可能解の数が多い問題では、選出領域を拡大して指向性交配の実行回数を増加させたときに解探索性能が改善されることがわかった。すなわち、最適化問題によって解探索性能を高める適切な選出領域は異なるため、一意に制御パラメータ  $S$  を決定することはできない。最適化問題の性質によらず、高い解探索性能を達成する適切な制御パラメータ  $S$  を解探索中に得ることを目的とした実験結果の解析から、指向性交配の実行率が約 90% になるように選出領域を制御する場合に、解探索性能が最大化することがわかった。この結果をもとに、選出領域を適応的に制御する方法を提案し、パラメータ  $S$  のチューニングを必要とせず、一度の実行で選出領域制御による解探索の改善効果が得られることを明らかにした。

### TNSDM-A : 有用な実行不可能解のアーカイブ法

TNSDM の世代ごとに解集団から消失してしまう有用な実行不可能解を解集団の中に維持して、繰り返し子生成に活用するための有用な実行不可能解のアーカイブ法を提案した。この方法を TNSDM に組み込んだアルゴリズムを TNSDM-A とし、テスト問題を用いて解探索性能を検証した結果、有用な実行不可能解をアーカイブ集団に保持して、繰り返し子生成に活用することによって、指向性交配による解探索性能の改善効果が高まることがわかった。また、アーカイブする有用な実行不可能解の最適個数は、選出領域内に存在する解の多さによって変化することがわかった。

### TNSDM-CSA : TNSDM-CS + TNSDM-A

有用な実行不可能解の選出領域制御法と、アーカイブ法の両方を TNSDM に組み込んだ TNSDM-CSA を提案した。6 章において、有用な実行不可能解の選出領域を縮小することによって解探索の指向性が高まり、解探索性能が改善される効果があることが示された。

しかし、選出領域が縮小されることによって、指向性交配の候補解集合が得られず、指向性交配を実行できないケースが生じるデメリットがあった。この問題に対して、アーカイブ集団を導入すると、選出領域を縮小しても指向性交配の候補解集合が得られやすくなるため、選出領域を縮小するデメリットを軽減できる。TNSDM-CSA の解探索性能を検証した結果、実行不可能解の選出領域制御法のみを用いる TNSDM-CS とアーカイブ法のみを用いる TNSDM-A より高い解探索性能を示すことが明らかになった。TNSDM-CSA は、解集団に有用な実行不可能解をアーカイブすることによって、有用な実行不可能解の選出領域に解が存在しやすくなるため、より選出領域を縮小した方法が、高い解探索性能を示すことがわかった。

### TNSDM-CG : 交叉遺伝子量の操作法

指向性交配で選択した実行可能解と有用な実行不可能解の交叉において、有用な実行不可能解から子に複製される遺伝子(変数)の量を操作する方法を提案した。実行不可能解から子に複製される遺伝子(変数)量を減少させると、生成される子は実行可能解にはなりやすいが、良い目的関数値を有する実行不可能解の親の遺伝子情報が得られにくくなる。逆に、実行不可能解から子に複製される遺伝子(変数)量を増加させると、生成される子は実行可能解にはなりやすいが、良い目的関数値を有する実行不可能解の親の遺伝子情報が得られやすくなる。この方法を TNSDM に組み込んだアルゴリズムを TNSDM-CG とし、テスト問題を用いて解探索性能を検証した結果、有用な実行不可能解から子に複製される遺伝子(変数)の量を操作することによって、指向性交配による解探索能力の改善効果が高まることが明らかになった。主として最適化する問題の実行可能率によって、有用な実行不可能解から子に複製される遺伝子(変数)の最適量は変化する。生成される子の実行可能解へのなりやすさと、実行不可能解の親から子に複製される有用な遺伝子情報の量のバランスによって、最適化性能を高めることが可能になる。

### TNSDM-CSACG : TNSDM-CS + TNSDM-A + TNSDM-CG

有用な実行不可能解の選出領域制御法とアーカイブ法、有用な実行不可能解から子に複製される遺伝子(変数)量の操作法を TNSDM に組み込んだ TNSDM-CSACG を提案した。9章において、特に問題の実行可能率が低い問題で、実行不可能解の親から子に複製される遺伝子量を減少させることで子の実行可能率を高め、解探索性能を改善することが示された。しかし、有用な実行不可能解のアーカイブ集団と選出領域制御と組み合わせることで、実行不可能解の質が高まり、組み合わせない場合と比べて子に複製する遺伝子量を増加させることで解探索性能が高まることがわかった。

## 11.2 今後の課題と展望

### 11.2.1 短期的課題

まず、短期的な課題としては、TNSDM-A と TNSDM-CG で必要となるパラメータの最適設定の指針や適応的制御法について検討する予定である。TNSDM-A において、最適なアーカイブサイズパラメータ  $\alpha^*$  は、第二の親候補解集合  $\mathcal{M}$  のサイズと関係があると考えられ、これを適応的制御の手掛かりにすることを検討している。また、TNSDM-CG における最適な遺伝子交換率パラメータ  $\gamma$  は、生成される子の実行可能率や、実行不可能解の親の制約違反量と実行可能解の親の制約許容量から推定できる可能性がある。適応制御が実現されれば、ユーザによるパラメータ設定が必要なくなるため、提案する TNSDM-A や-CG が利用しやすくなる。

### 11.2.2 長期的展望

次に、長期的な展望としては、工学設計最適化の課題解決へ取り組むことを計画している。進化計算は、新幹線 N700 系の先頭車両形状やジェット旅客機 MRJ の翼形状の最適化など、産業界の新しい設計手法として利用されはじめているが、いくつかの課題があり、その適用可能範囲が限られている。従来の進化計算は、新たな解の生成と評価を 10～100 万回ほど繰り返して最適化する。しかし、工学設計の最適化では、ひとつの解の評価に時間がかかる。そのため、膨大な評価回数を費やして最適化することは非現実的である。問題に制約条件がある場合、その条件を満たさない実行不可能解は、これまで単に捨てられてしまい、評価時間が無駄になっていた。これに対して、本研究で提案した指向性交配は、実行不可能解の情報を十分に活用し、解探索に利用できる。特に工学設計の最適化においては、評価シミュレーションの時間が無駄にならないばかりか解探索を促進する効果をもたらす。このように、特に工学設計を標的とした最適化に取り組むことは、本研究で提案した指向性交配の利点を活かすための意義だけでなく、進化計算に対する社会的期待に応える意義がある。

工学設計の最適化における従来の進化計算の課題は、第一に最適化時間の短縮、第二に製造誤差への対処第三に意思決定支援強化にある。第一の課題に対しては、本研究の成果を基礎とし、(A) 指向性交配において実行可能解になりやすく目的関数値の良い子を生成可能にすること、(B) 評価関数を推定して解探索に活用すること、(C) 解の生成評価を効果的に並列化することが考えられる。第二の課題に対しては、(D) 生じる製造誤差に対して解の実行可能性を考慮可能にすること、第三の課題に対しては、(E) 最適解の変数、目的関数値情報だけでなく、制約条件も考慮した意思決定支援があげられる。

### (A) 効率的な子個体生成

現在の指向性交配では、目的関数値を考慮して実行可能解の親の目的関数値を高める可能性のある実行不可能解を選出し、交配に活用することによって解探索性能が向上することがわかっている。しかし、目的関数値以外の情報は使用せず、子は選出された親の変数をランダムに組み合わせることによって生成している。これに対して、解の持つ制約違反量や、実行可能解の親とその交配候補解集合  $M$  に含まれる複数の解の変数情報をもとに、より目的関数値を高めやすく実行可能解になりやすい解の変数を推定し、子生成に活用することによって解探索効率を高め、最適化時間を短縮できる可能性がある。

### (B) 評価関数の推定と活用

高い評価値を持つ実行不可能解は、実行可能解の評価値を高めるための手掛かりになるが、工学設計の最適化の場合、一つひとつの解の評価に時間を要するため膨大な数の解を評価できず、解探索の手掛かりにできる実行不可能解が少なくなる。そのため、制約目的関数を推測し、解探索の手掛かりとなりうる仮想解を挿入して指向性交配する方法などが考えられる。

### (C) 解の生成評価の並列化

解の並列評価による単純な最適化時間の短縮に加えて、高い評価値を示す実行不可能解が存在する実行可能解へ優先的に計算資源を割り当てて子を生成・評価するなど、最適化アルゴリズム上の並列化への工夫も考えられる。

### (D) 製造誤差の考慮

工学設計の最適化では、結果として得られた設計案(解)に寸分の違いもなく製造することは困難である。そのため、設計案が実行可能解であっても、製造誤差が原因で制約違反が生じる恐れがある。製造誤差に対する解の実行可能性も考慮して最適化することで、実行可能性を考慮した設計の取捨選択や意思決定が可能になると考えられる。

### (E) 制約を考慮した意思決定支援

進化計算による多目的最適化の結果としては、パレートフロントを近似する複数の解が獲得できる。それらの解から、目的間のトレードオフ関係や目的関数値と変数の関係などの知識を得ながら、最終的に意思決定者は好ましい一つの解を選択することになる。意思決定支援のため、これまでに、多次元の目的関数空間を視覚的に提示するための研究 [52, 53] や、選好情報をもとにパレートフロントの一部のみを探索し、局所的な最適性を高める研究 [54, 55, 56] などが行われている。これらは特に目的関数に注目した手法であ



る。工学設計などの実問題においては、目的関数のみでなく、最適解の制約条件との関係も意思決定に関与する可能性がある。制約条件をどの程度余裕をもって満足しているかや、制約条件を緩めるとどれくらい目的関数値を高められるかなどの情報を提示することによって意思決定の幅が広がる。そのため、わかりやすくこれらを示す方法の構築が必要であると考える。

このように、今後、工学設計を標的とした最適化の研究に取り組むことで、これまでの研究を学術的に発展させるとともに、進化計算に対する社会的期待に応えたいと考えている。

# 謝辞

本論文をまとめるにあたり始終多大なるご指導と御教示をいただいた高玉圭樹教授，佐藤寛之助教，指導教員の西野哲朗教授，高橋治久教授に心より感謝の意を表します。博士論文の審査をしていただいた庄野逸教授，高橋裕樹准教授，特別研究員制度によって支援いただいた日本学術振興会様，3段式ハイブリッドロケットの概念設計最適化問題の評価モジュールを提供下さった首都大学東京の金崎雅博准教授，金森文男様，並びに研究室の人々に心より感謝申し上げます。また，長い学生生活を支えてくれた家族にも感謝しています。

# 参考文献

- [1] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.
- [2] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, Kluwer Academic Publishers, 2002.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [4] 北野 宏明, 遺伝的アルゴリズム, 産業図書, 1993.
- [5] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.
- [6] H. P. Schwefel, *Evolution and Optimum Seeking*, Wiley & Sons, 1995.
- [7] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, Vol. 11, pp. 341–359, 1997.
- [8] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” *Proc. of IEEE International Conference on Neural Networks IV*. pp. 1942–1948, 1995.
- [9] M. Dorigo, “Optimization, Learning and Natural Algorithms,” *Ph.D. Thesis*, Politecnico di Milano, Italy, 1992.
- [10] C. A. C. Coello and G. B. Lamont, *Applications of Multi-Objective Evolutionary Algorithms*, World Scientific, Singapore, 2004.
- [11] 成瀬 功, 鳥居 昭彦, “次世代新幹線車両 (N700) の先頭形状の開発,” *JREA*, Vol. 47 (7), pp. 30199–30202, 2004.
- [12] K. Chiba, S. Obayashi, K. Nakahashi, and H. Morino, “High-Fidelity Multidisciplinary Design Optimization of Wing Shape for Regional Jet Aircraft,” *Proc. of 2005 Evolutionary Multi-Criterion Optimization (EMO2005)*, LNCS (Springer), Vol. 3410, pp. 621–635, 2005.
- [13] E. M. Montes, *Constraint-Handling in Evolutionary Optimization*, Springer, 2009.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast Elitist Multi-Objective Ge-

- netic Algorithm: NSGA-II,” *IEEE Trans. on Evolutionary Computation*, Vol. 6, pp. 182–197, 2002.
- [15] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer, 2004.
- [16] N. Srinivas and K. Deb, “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evolutionary Computation*, Vol. 2 (3), pp. 221–248, 1994.
- [17] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, “GA-Based Decision Support System for Multicriteria Optimization,” *Proc of the Int’l Conf. on Systems, Man, and Cybernetics*, Vol. 2, pp. 1556–1561, 1995.
- [18] A. Osyczka and S. Kundu, “A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm,” *Structural Optimization*, Vol. 10 (2), pp. 94–99, 1995.
- [19] M. Miyakawa, K. Takadama, and H. Sato, “Controlling Selection Areas of Useful Infeasible Solutions for Directed Mating in Evolutionary Constrained Multi-Objective Optimization,” *Annals of Mathematics and Artificial Intelligence*, Springer, DOI: 10.1007/s10472-015-9455-9, 2015.
- [20] Y. Kitagawa, M. Nakamiya, M. Kanazaki, and T. Shimada, “Multi-Stage Hybrid Rocket Conceptual Design for Micro-Satellites Launch using Genetic Algorithm,” *Trans. of JSASS*, Vol. 55 (4), 2012.
- [21] T. Ray, K. Tai, and C. Seow, “An evolutionary algorithm for multiobjective optimization,” *Eng. Optim.*, Vol. 33 (3), pp. 399–424, 2001.
- [22] H. Sato, H. Aguirre, and K. Tanaka, “Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs,” *Proc. 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO2007)*, Lecture Note in Computer Science (Springer), Vol. 4403, pp. 5–20, 2007.
- [23] F. Hoffmeister and J. Sprave, “Problem-independent handling of constraints by use of metric penalty functions,” *Proc. of the 5th Annual Conference on Evolutionary Programming (EP 1996)*, pp. 289–294, 1996.
- [24] T. Bäck, F. Hoffmeister, and H. Schwefel, “A Survey of Evolution Strategies,” *Proc. of the 4th International Conference on Genetic Algorithms*, pp. 2–9, 1991.
- [25] C. A. C. Coello and A. D. Christiansen, “MOSES: A Multiobjective Optimization Tool for Engineering Design,” *Engineering Optimization*, Vol. 31 (3), pp. 337–368, 1999.
- [26] E. Zitzler and L. Thiele, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” *IEEE Trans. on Evolutionary Computation*, Vol. 3 (4), pp. 257–271, 1999.
- [27] A. Jaskiewicz, “On the performance of multiple-objective genetic local search on the

- 0/1 knapsack problem - A comparative experiment,” *IEEE Trans. on Evolutionary Computation*, Vol. 6 (4), pp. 402–412, 2002.
- [28] H. Ishibuchi and S. Kaige, “Effects of Repair Procedures on the Performance of EMO Algorithms for Multiobjective 0/1 Knapsack Problems,” *Proc. of the 2003 Congress on Evolutionary Computation (CEC’2003)*, Vol. 4, pp. 2254–2261, 2003.
- [29] A. Homaifar, S. H. Y. Lai, and X. Qi, “Constrained Optimization via Genetic Algorithms,” *Trans. of The Society for Modeling and Simulation International - SIMULATION*, Vol. 62 (4), pp. 242–254, 1994.
- [30] J. Joines and C. Houck, “On the Use of Non-Stationary Penalty Functions to Solve Non-linear Constrained Optimization Problems with Gas,” *Proc. of the First IEEE Conference on Evolutionary Computation*, pp. 579–584, 1994.
- [31] R. Farmani and J. A. Wright, “Self-Adaptive Fitness Formulation for Constrained Optimization,” *IEEE Trans. on Evolutionary Computation*, Vol. 7 (5), pp. 445–455, 2003.
- [32] N. Srinivas and K. Deb, “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms,” *Evolutionary Computation Journal*, Vol. 2 (3), pp. 221–248, 1994.
- [33] K. Deb, “Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design,” *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pp. 135–161, John Wiley & Sons, 1999.
- [34] J. Hazra and A. K. Sinha, “A multi-objective optimal power flow using particle swarm optimization,” *European Trans. on Electrical Power*, Vol. 21 (1), pp. 1028–1045, 2011.
- [35] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, “Constraint Handling in Multiobjective Evolutionary Optimization,” *IEEE Trans. on Evolutionary Computation*, Vol. 13 (3), pp. 514–525, 2009.
- [36] E. M. Montes and C. A. C. Coello, “Constrained Optimization via Multiobjective Evolutionary Algorithms,” *Multiobjective Problem Solving from Nature, Part I*, Springer, pp. 53–75, 2008.
- [37] T. T. Binh and U. Korn, “MOBES: A multi-objective evolution strategy for constrained optimization problems,” *Proc. of Int. Conf. Genetic Algorithms*, pp. 176–182, East Lansing, MI, 1997.
- [38] S. Kukkonen and J. Lampinen, “Constrained Real-Parameter Optimization with Generalized Differential Evolution,” *Proc. of 2006 IEEE Congress on Evolutionary Computation (CEC2006)*, pp. 911–918, 2006.
- [39] A. Oyama, K. Shimoyama, and K. Fujii, “New Constraint-Handling Method for Multi-Objective and Multi-Constraint Evolutionary Optimization,” *Trans. of the Japan Society for Aeronautical and Space Sciences*, Vol. 50 (167), pp. 56–62, 2007.

- [40] Q. Zhang and H. Li, “MOEA/D: A Multi-objective evolutionary algorithm based on decomposition,” *IEEE Trans. on Evolutionary Computation*, Vol. 11 (6), pp. 712–731, 2007.
- [41] H. Jain and K. Deb, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach,” *IEEE Trans. on Evolutionary Computation*, Vol. 18 (4), pp. 602–622, 2014.
- [42] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1999.
- [43] J. Knowles and D. Corne, “On Metrics for Comparing Non-dominated Sets”, *Proc. 2002 IEEE Congress on Evolutionary Computation*, pp.711–716, 2002.
- [44] M. Sato, H. Aguirre, and K. Tanaka, “Effects of  $\delta$ -Similar Elimination and Controlled Elitism in the NSGA-II Multiobjective Evolutionary Algorithm,” *Proc. of 2006 IEEE Congress on Evolutionary Computation (CEC2006)*, pp. 1164–1171, 2006.
- [45] 柳浦睦憲, “組合せ最適化の数理: 計算困難問題への挑戦”, *数理科学*, 第 40 卷 12 号, No. 474, pp. 21–27, 2002.
- [46] H. Sato, “Analysis of Inverted PBI and Comparison with Other Scalarizing Functions in Decomposition Based MOEAs,” *Journal of Heuristics*, Springer, Vol. 21 (6), pp. 819–849, 2015.
- [47] E. Zitzler and M. Laumanns, “Test Problem Suite,” <http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/> (Accessed 24 March 2016).
- [48] G. Syswerda, “Uniform Crossover in Genetic Algorithms,” *Proc. of ICGA89*, pp. 2–9, 1989.
- [49] K. Deb and M. Goyal, “A Combined Genetic Adaptive Search (GeneAS) for Engineering Design,” *Computer Science and Informatics*, Vol. 26 (4), pp. 30–45, 1996.
- [50] H. Sato, H. Aguirre, and K. Tanaka, “Variable Space Diversity, Crossover and Mutation in MOEA Solving Many-objective Knapsack Problems,” *Annals of Mathematics and Artificial Intelligence*, Springer, Vol. 68 (4), pp. 197–224, 2013.
- [51] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable Test Problems for Evolutionary Multi-Objective Optimization,” *TIK-Report No.112*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.
- [52] D. J. Walker, R.M. Everson, and J.E. Fieldsend, “Visualizing mutually nondominating solution sets in many-objective optimization,” *IEEE Trans. on EC*, Vol. 17 (2), pp. 165–184, 2013.
- [53] T. Tusar and B. Filipic, “Visualization of Pareto front approximations in evolutionary

- multiobjective optimization: a critical review and the prosection method,” *IEEE Trans. on EC*, (early access)
- [54] K. Deb and J. Sundar, “Preference point based multi-objective optimization using evolutionary algorithms,” *Proc. of 2006 Genetic and Evolutionary Computation Conference (GECCO 2006)*, pp. 635–642, 2006.
- [55] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, “Articulating user preferences in many-objective problems by sampling the weighted hypervolume,” *Proc. of 2009 Genetic and Evolutionary Computation Conference (GECCO 2009)*, pp. 555–562, 2009.
- [56] K. Narukawa, Y. Setoguchi, Y. Tanigaki, M. Olhofer, B. Sendhoff, and H. Ishibuchi, “Preference representation using Gaussian functions on a hyperplane in evolutionary multi-objective optimization,” *Soft Computing*, Online first, pp. 1–25, 2015.

# 関連発表

## 学会誌論文

1. 宮川 みなみ, 佐藤寛之, “二段階の非支配ソーティングと指向性交配による制約付き多目的最適化”, 進化計算学会論文誌, Vol. 3, No. 3, pp. 185–196, 2013.
2. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Archive of Useful Solutions for Directed Mating in Evolutionary Constrained Multiobjective Optimization,” The Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol. 18, No. 2, pp. 221–231, 2014.
3. Kouhei Tomita, Minami Miyakawa, and Hiroyuki Sato, “Adaptive Control of Dominance Area of Solutions in Evolutionary Many-Objective Optimization,” New Mathematics and Natural Computation Vol. 11, No. 2, pp. 135–150, 2015.
4. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Controlling Selection Areas of Useful Infeasible Solutions for Directed Mating in Evolutionary Constrained Multi-Objective Optimization”, Annals of Mathematics and Artificial Intelligence, Springer, DOI: 10.1007/s10472-015-9455-9, 2015. (Online First)

## 国際会議論文

1. Minami Miyakawa and Hiroyuki Sato, “Effects of Two-Stage Non-Dominated Sorting and Directed Mating in Constrained MOEAs,” Proc. of 2012 International Workshop on Modern Science and Technology (IWMST 2012), pp. 372–377, 2012.
2. Minami Miyakawa and Hiroyuki Sato, “An Evolutionary Algorithm using Two-stage Non-dominated Sorting and Directed Mating for Constrained Multi-objective Optimization,” The 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems (SCIS-ISIS2012), pp. 1441–1446, 2012.
3. Kouhei Tomita, Minami Miyakawa and Hiroyuki Sato, “Adaptive Control of Dominance Area in MOEAs on Many-Objectives Concave Problems,” The 16th Asia Pacific Symposium of Intelligent and Evolutionary Systems (IES2012), pp. 8–14, 2012.



4. Hiroyuki Sato, Minoru Imajima, and Minami Miyakawa, “A Study on Evolutionary Multi-Objective Racing Car Design Optimization using a Parallel MOEA/D,” Proc. of the 2013 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP2013), pp. 608–611, 2013.
5. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Two-Stage Non-Dominated Sorting and Directed Mating for Solving Problems with Multi-Objectives and Constraints,” Proc. of 2013 Genetic and Evolutionary Computation Conference (GECCO 2013), pp. 647–654, 2013.
6. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Controlling Selection Area of Useful Infeasible Solutions in Directed Mating for Evolutionary Constrained Multi-objective Optimization,” Proc. of the 8th Learning and Intelligent Optimization Conference (LION 8), LNCS 8426, pp. 137–152, 2014.
7. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Controlling Selection Area of Useful Infeasible Solutions and Their Archive for Directed Mating in Evolutionary Constrained Multiobjective Optimization,” Proc. of 2014 Genetic and Evolutionary Computation Conference (GECCO 2014), pp. 629–636, 2014.
8. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Directed Mating Using Inverted PBI Function for Constrained Multi-Objective Optimization,” Proc. of 2015 IEEE Congress on Evolutionary Computation (CEC2015), pp. 2929–2936, 2015.
9. Hiroyuki Sato, Minami Miyakawa, Elizabeth Perez-Cortes, “An Efficient Parallel Solution Evaluations in MOEA/D by Avoiding Overlaps of Neighbors,” Proc. of Workshop on Evolutionary Multi-Objective Optimization at 2015 IEEE Congress on Evolutionary Computation (CEC2015), pp. 11–16, 2015.
10. Tomohisa Hashimoto, Hiroyuki Sato, Minami Miyakawa, “A Study on Gradual Enhancement of the Approximation Granularity of Pareto Front in Evolutionary Many-Objective Optimization,” Proc. of Workshop on Evolutionary Multi-Objective Optimization at 2015 IEEE Congress on Evolutionary Computation (CEC2015), pp. 17–22, 2015.
11. Hiroyuki Sato, Minami Miyakawa, Elizabeth Perez-Cortes, “A Parallel MOEA/D Generating Solutions in Minimum Overlapped Update Ranges of Solutions,” Proc. of Late-Breaking Abstracts at 2014 Genetic and Evolutionary Computation Conference (GECCO 2015), pp. 775–776, 2015.
12. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Control of Crossed Genes Ratio for Directed Mating in Evolutionary Constrained Multi-Objective Optimization,” Proc. of Student Workshop at 2015 Genetic and Evolutionary Computation

- Conference (GECCO 2015), pp. 1201–1204, 2015.
13. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “Control of Variable Exchange Probability for Directed Mating in Evolutionary Constrained Multi-Objective Continuous Optimization,” The 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015), pp. 33–38, 2015.
  14. Hiroyuki Sato, Kouhei Tomita, and Minami Miyakawa, “Preferred Region Based Evolutionary Multi-Objective Optimization Using Parallel Coordinates Interface,” The 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015), pp. 89–94, 2015.
  15. Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “A Study on Directional Repair of Infeasible Solutions for Multi-Objective Knapsack Problems,” Proc. of the 2016 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP2016), Mar. 2016. (to appear)
  16. Tomohisa Hashimoto, Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato, “A Study on Evolutionary Multi-level Robust Solution Search for Multi-objective Optimization Involving Multi-dimensional Noise,” Proc. of the 2016 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP2016), Mar. 2016. (to appear)

## 国内学会発表

1. 宮川 みなみ, 佐藤 寛之, 服部 聖彦, 高玉 圭樹, “ $m$  目的  $k$  ナップザック問題における実行不可能解の修復法の検討”, 電子情報通信学会総合大会, p. 73, 2011.
2. 宮川 みなみ, 佐藤 寛之, 服部 聖彦, 高玉 圭樹, “解の支配領域制御法を用いた MOEA による制約付き多目的最適化問題の解法に関する一検討”, 平成 23 年度電子情報通信学会信越支部大会, p. 154, 2011.
3. 宮川 みなみ, 佐藤 寛之, “複数制約付き多目的最適化における指向性交配のペア選択に関する検討”, 進化計算学会 進化計算シンポジウム 2012, pp. 431–438, 2012.
4. 冨田 浩平, 宮川 みなみ, 佐藤 寛之, “MOEA における解の支配領域の適応的制御法とパレート形状に対する頑健性”, 進化計算学会 進化計算シンポジウム 2012, pp. 416–423, 2012.
5. 宮川 みなみ, 佐藤 寛之, “複数制約付き多目的解探索を促進する実行不可能解のアーカイブに関する基礎検討”, 第五回進化計算学会研究会, pp. 88–93, 2013.
6. 佐藤 寛之, 佐藤 圭二, 宮川 みなみ, Elizabeth Perez-Cortes, 高玉 圭樹, “分解に基づく進化型多目的最適化法の効果的な並列化に関する検討”, 計測自動制御学会 システム・情報部門 学術講演会 2013, pp. 599–603, 2013.

7. 宮川 みなみ, 高玉 圭樹, 佐藤 寛之, “指向性交配における有用な実行不可能解の選出領域制御に関する検討”, 進化計算学会 進化計算シンポジウム 2013, pp. 22–28, 2013.
8. 宮川 みなみ, 佐藤 寛之, 金森 文男, 金崎 雅博, “3 段式ハイブリッドロケットの概念設計最適化における実行不可能解の有効活用に関する一検討”, 日本機械学会 第 27 回計算力学講演会, in CD-ROM, 2014.
9. 宮川 みなみ, 高玉 圭樹, 佐藤 寛之, “制約付き多数目的最適化のためのリファレンスラインを用いた指向性交配の検討”, 進化計算学会 進化計算シンポジウム 2014, pp. 363–370, 2014.
10. 橋本 知尚, 宮川 みなみ, 佐藤 寛之, “ノイズを有する多目的最適化問題におけるマルチレベルのロバスト選好解探索法の検討”, 進化計算学会 進化計算シンポジウム 2014, pp. 26–33, 2014.
11. 佐藤 寛之, 富田 浩平, 宮川 みなみ, “平行座標ユーザインターフェースを用いた多数目的選好解集合探索に関する一検討”, 第 8 回進化計算学会研究会, pp. 100–106, 2015.
12. 宮川 みなみ, 高玉 圭樹, 佐藤 寛之, “制約付き多目的最適化のための指向性交配における解の選出領域の適応制御に関する一検討”, 進化計算学会 進化計算シンポジウム 2015, pp. 255–261, 2015.
13. 中川 智, 宮川 みなみ, 高玉 圭樹, 佐藤 寛之, “補助重みベクトル群による多数目的最適化の促進に関する一検討”, 進化計算学会 進化計算シンポジウム 2015, pp. 226–232, 2015.
14. 橋本 知尚, 宮川 みなみ, 高玉 圭樹, 佐藤 寛之, “多次元ノイズを含む多目的最適化におけるスカラー化関数に基づくロバスト解探索”, 進化計算学会 進化計算シンポジウム 2015, pp. 241–247, 2015.