協調型共進化 GA による動的施設配置問題の一解法*

張 明†・大倉 和博‡・上田 完次‡

A Cooperatively Coevolutionary Genetic Algorithm
to the Dynamic Facility Layout Problems*

Ming CHANG†, Kazuhiro OHKURA‡ and Kanji UEDA‡

A cooperative coevolutionary genetic algorithm (CCGA) to the dynamic facility layout problems (DFLP) is examined. The CCGA has two major features. Firstly, a multi-population approach is adopted. The entire solution is constructed by individuals picked up from each population which corresponds to layouts of a production period in DFLP. An individual's fitness value is calculated with the best individuals of the other populations so that the populations can co-evolve in a cooperative manner. Secondly, steady-state selection process is embedded in each population to promote the exploitation in the genetic search. Computer simulations on two benchmark problems are conducted. It is found that the CCGA outperforms the other GAs consistently.

1. はじめに

施設配置問題 (Facility Layout Problems: FLP) は、複数の物理的施設の配置に関する問題である。その定式化によって、地域内の複数の倉庫間の配達業務問題、工場内の複数の機械配置問題や分散メモリ型マルチプロセッサにおける配線問題などに適用できる [12]。

FLP には、量的目的関数と質的目的関数の二つの目的関数が定義されている。量的目的関数は原料などの運搬コストの最小化を表す一方、質的目的関数は施設間の距離の最小化を表す [17]。

現実の工場においては、原料運搬などのコストが、工場の全稼働コストの 20%~50% を占め、効率的な設備配置はそのコストを 10%~30% 削減でき、生産性の向上を導くといわれており [18]、そのため、FLP が重要な問題となる。

FLP には静的 FLP と動的 FLP がある。これまで、FLP に関する多くの研究は特に静的 FLP (Static Facility

Layout Problem: SFLP) に焦点を当てられてきた [12]。SFLP は Quadratic Assignment Problem (QAP) によって定式化できることが知られている [11]。しかし、このような定式化においては、問題解決の所要時間は、施設数の増大に伴い指数的に増加してしまう [9]。動的 FLP (Dynamic Facility Layout Problem: DFLP) は近年、多品種・少量生産を念頭において、注目されている。DFLP は、各生産期間における施設間の原料運搬量の変動に応じて施設配置を変えることで、生産コストの削減をしようとするものである。DFLP を難しくするのは、施設の再配置にコストが生じるためである。SFLP ではある最適な配置が求められるが、DFLP は全生産期間を通じての最適な配置が求められる。DFLP におけるコストは各生産期間で行われる施設の再配置によって生じるコストと施設間の原料運搬コストの合計である。

DFLP に対して、Dynamic Programming (DP), Tabu-Search などの手法が提案されてきたが [3]、大規模な DFLP に対しては、より効率的な最適手法が必要となる。近年、遺伝的アルゴリズム (GA) による DFLP の解決が試みられている [4,6]。GA は生物の遺伝と進化を計算機上で模倣し人工物の環境適応、学習および最適化に適用する手法であり、多峰性、非線形や制約付き目的関数を持つ問題に特に適するといわれている。DP や Tabu-Search などの最適手法における「解個体」による探索に対して、GA は、「解集団」による探索が特徴とい

* 原稿受付 2001 年 2 月 15 日

† 岐阜県生産情報技術研究所 Gifu Prefecture Research Institute of Manufacturing Information Technology; 19-179-4 Sue, Kakamigahara city, Gifu 509-0108, JAPAN

‡ 神戸大学 工学部 Faculty of Engineering, Kobe University; Rokkodai, Nada-ku, Kobe city, Hyogo 657-8501, JAPAN

Key Words: genetic algorithms, cooperative coevolution, dynamic facility layout problem.

える。そのため、効率的な探索には、探索集団の多様性維持が不可欠である。これまで、個体が問題の完全解をコーディングし、適応度に基づいて選ばれた個体の突然変異や個体間の組換えによって新しい個体を生成するアプローチがしばしばとられている [4,6]。しかし、単純な組換え操作では実行不可能な解が生成されることがあるため、付加的な操作が必要になる。そのため、多くのGAによる研究は効率的な組換え操作の構築に着目している [1,4,6]。

本論文では、協調型共進化GAによるDFLPへのアプローチについて検討する。DFLPにおける各生産期間における施設配置間の協調的な側面に注目し、協調型共進化GAを適用するにあたり、DFLPを各生産期間に分割し、各生産期間の配置に一つの探索集団を対応させ、集団内の個体が対応する生産期間での配置をコーディングする。DFLPの解は各生産期間の配置をコーディングしている個体らによって構成され、個体はその解のよさによって評価される。以下では、FLPを定式化した後、提案するアルゴリズムを説明し、計算機実験の結果を示す。

2. 施設配置問題

施設配置問題 (Facility Layout Problems: FLP) は、 m 個の施設を n ($m \leq n$) ヶ所に配置し、固定投資と運搬コストの合計を最少にする問題である。これまでの多くの研究は、静的施設配置問題に焦点を当ててきた。ここで静的とは、施設間の原料運搬量が生産中に固定していることをいう。

2.1 静的施設配置問題

静的施設配置問題 (Static Facility Layout Problem: SFLP) は Quadratic Assignment Problem (QAP) として定式化できる [11]。QAPにおいては、目的関数は2次関数、制約条件は線形関数で表される [12]。QAPの定式化においては、配置場所の数と配置される施設の数は等しい。しかし、架空の配置場所や施設を導入することで、すべての問題はQAPによって定式化できることが知られている [5,10]。以下は参考文献 [17] によるSFLPの定式化を示す。

The SFLP Model

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N a_{ijkl} X_{ij} X_{kl} \quad (1)$$

subject to

$$\sum_{i=1}^N X_{ij} = 1, \quad j = 1, \dots, N \quad (2)$$

$$\sum_{j=1}^N X_{ij} = 1, \quad i = 1, \dots, N \quad (3)$$

$$X_{ij} \in \{0,1\}, \quad i, j = 1, 2, \dots, N \quad (4)$$

where:

$$X_{ij} = \begin{cases} 1 & \text{施設 } i \text{ が場所 } j \text{ に配置される} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$a_{ijkl} = \begin{cases} f_{ik} d_{jl} & \text{if } i \neq k \text{ or } j \neq l \\ c_{ij} & \text{if } i = k \text{ and } j = l \end{cases} \quad (6)$$

N : 施設 (場所) 数

i, k : i, k 番目の施設

j, l : j, l 番目の場所

c_{ij} : 場所 i での施設 j の固定投資

d_{jl} : 場所 j と l の間の距離、あるいは運搬コスト; $d_{jj} = 0$ とする

f_{ik} : 施設 i と k の間の原料運搬量

c_{ij} 項が0あるいは、すべての i, j について等しい場合、(6)式は以下のように書き換えられる。

$$a_{ijkl} = f_{ik} d_{jl} \quad (7)$$

本論文では、この式に従う。

2.2 動的施設配置問題

動的施設配置最適化問題 (Dynamic Facility Layout Problem: DFLP) は、SFLPを拡張した形で参考文献 [17] で紹介された。以下は、参考文献 [3] によるDFLPの定式化を示す。

The DFLP Model

$$\text{Min} \left\{ \sum_{t=2}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{tijl} Y_{tijl} + \sum_{t=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N C_{tijkl} X_{tij} X_{tkl} \right\} \quad (8)$$

subject to

$$\sum_{i=1}^N X_{tij} = 1, \quad j = 1, \dots, N \quad t = 1, \dots, P \quad (9)$$

$$\sum_{j=1}^N X_{tij} = 1, \quad i = 1, \dots, N \quad t = 1, \dots, P \quad (10)$$

$$X_{tij} \in \{0,1\}, \quad i, j = 1, 2, \dots, N \quad t = 1, \dots, P \quad (11)$$

where:

$$X_{tij} = \begin{cases} 1 & \text{生産期間 } t \text{ に,} \\ & \text{施設 } i \text{ が場所 } j \text{ に配置される} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$C_{tijkl} = f_{tik} d_{jl} \quad (13)$$

$$Y_{tijl} = X_{(t-1)ij} \times X_{til} \quad (14)$$

- P : 考慮する生産期間数
 N : 施設 (場所) 数
 i, k : i, k 番目の施設
 j, l : j, l 番目の場所
 Y_{tijl} : 生産期間 t に、施設 i を場所 j から l に再配置するかどうかに関する 0,1 の 2 値変数, 1 は再配置を表す
 A_{tijl} : 生産期間 t に、施設 i を場所 j から l に再配置するコスト
 C_{tijk} : 生産期間 t に、場所 j での施設 i と場所 l での施設 k の間の原料運搬コスト
 d_{jl} : 場所 j と l の間の距離, あるいは運搬コスト; $d_{jj} = 0$ とする
 f_{tik} : 生産期間 t に、施設 i と k の間の原料運搬量

2.3 制約付き動的施設配置問題

前述の DFLP では再配置に使われる資金に関する考慮はなかった。しかし、現実には、これは重要な問題であり、これを制約条件の一つとして加えたものが制約付き動的施設配置問題 (Constrained Dynamic Facility Layout Problem: CDFLP) である [2]。以下にその定式化を与える。

$$\sum_{t=2}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{tijl} Y_{tijl} \leq B \quad (15)$$

(15) 式によって、再配置に使える資金は B 以下に制限されている。

また、再配置に使える全資金のかわりに、各生産期間における再配置に使える資金に制約を加える CDFLP は参考文献 [6] によって定式化された。本論文では以下のように定式する。

$$\sum_{t=2}^T (B_t - \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{tijl} Y_{tijl}) \leq 0 \quad (16)$$

where

$$T = 2, \dots, P$$

B_t : 生産期間 t での再配置投資資金に関する制約

(16) 式は再配置が、それに使える資金が蓄えられた時だけ行えることを表す。また、各生産期間における再配置投資資金を $(B_2, B_3, \dots, B_P)$ のように表す。

3. 協調型共進化 GA によるアプローチ

3.1 協調型共進化 GA

DFLP に対して、これまでに DP, Tabu-Search, GA など、様々な手法が提案されてきたが [3]、中でも、GA によるアプローチが良い結果を出している [4,6]。しかし、問題の数だけ GA のバリエーションがあるといわれるように、GA の問題依存性も多くの研究によって指摘され

ている。そのため、これまで問題ごとに *Canonical GA* (CGA) を改良して適用されてきた。

協調型共進化 GA は CGA と二つの点において異っている。(1) 多集団を使って解候補を表現し、各集団内の個体は部分解を表す。個体の評価は他集団中の個体と生成した完全な問題解の良さによって行われる。(2) 各集団内には Steady-State Reproduction [7] を用いる。

これまで、関数最適化 [14], Artificial Neural Network の最適化 [15] とロボット制御 [8,16] などの問題に対して、協調型共進化 GA が CGA より良い性能を示すことが報告されている。これらの研究によって、協調型共進化 GA の問題点も明らかになってきた。(1) 問題分割に関して、一般的な基準はなく、問題の具体的構造によって異なる。(2) 個体の適応度の決め方に関して、*classifier system* での信頼度割り当て問題と同じ困難に直面する。すなわち、解は個体によって生成されたもので、解 (協調している個体) は明確な評価関数さえ存在すれば、容易に評価できるが、各個体が生成された解にどれぐらい貢献しているかは解の内部構造を分析しないかぎり評価できない。そのため、協調型共進化 GA を具体的問題の特性に合わせて設計する必要がある。従来の共進化 GA に関する研究は主に進化過程のダイナミクスに注目しているもので、共進化 GA の問題解決性能に対する検討は十分に行わなかった。共進化 GA と CGA との性能比較研究 [14] はあるが、工学問題における共進化 GA の性能検証が少ない。そのため、本論文は共進化 GA の問題解決能力をすでに多くの GA 手法によってチャレンジされてきた動的施設配置問題を用いて検討する。

協調型共進化 GA を P 生産期間の N 施設の DFLP に適用するにあたり、まず、DFLP を各生産期間に分割する。第 t 集団内の個体は、生産期間 t での施設配置を表す。そのため、集団の数は P と等しい。各個体の対する初期評価は他集団内のランダムな個体と生成した完全解によって行われる。

3.1.1 個体表現

FLP においては、最も便利な個体表現方法は、集合 $\{1, 2, \dots, N\}$ の順列とする。第 $t(t \in \{1, 2, \dots, P\})$ 集団内の第 k 個体を a_{tk} と表す。

3.1.2 個体の評価

協調型共進化 GA では、部分解を表している各個体はほかの集団内の最適個体と構成した完全解によって評価される¹。 f を適応度関数、 a_t^* を第 t 集団 ($t \in \{1, 2, \dots, P\}$) 内の最適個体とし、第 i 集団内の個体 a_{ik} を評価する場合、まず a_{ik} とほかの集団内の最適個体 a_t^* ($t \in \{1, 2, \dots, P\} \wedge i \neq t$) を用い、完全解 $(a_1^*, \dots, a_{i-1}^*, a_{ik}, a_{i+1}^*, \dots, a_P^*)$ を構成

¹初期集団における個体評価は他集団内のランダムに選ばれた個体と生成した完全解によって行われる。そして、次の世代から、集団内の最も高い適応度を持つ個体を最適個体とする。

し、その適応度 $f(a_1^*, \dots, a_{i-1}^*, a_{i_k}, a_{i+1}^*, \dots, a_p^*)$ を a_{i_k} の適応度とする。この評価方法は、各部分問題が独立な場合に適するが、部分問題間に相関のある場合にも適するとはいえないが、その簡潔性のため、この評価方法を用いる。

3.1.3 The Steady-State Reproduction

初期集団はランダムに生成され、評価の後、集団の更新は適応度の高い個体によるあるランダムに選ばれた個体への置換によって行われる。たとえば、集団 t において、 a_t^* をその集団の最適解、 a_{t_χ} を集団からランダムに選ばれた個体とし、集団 t の更新は、 $a_{t_\chi} = a_t^*$ によって行われる。 χ は $\{1, 2, \dots, N\}$ に一様分布する整数乱数とする。

3.1.4 遺伝的操作

各集団内には、組換え操作は使わない。突然変異には、swap mutation operator、つまり、施設配置を表す数列中の二つのランダムに選ばれた施設の位置を交換する操作を使用した。 $a_{t_k} = (\dots, m_{\chi_1}, \dots, m_{\chi_2}, \dots)$ を集団 t 内の k 番目個体とし、 a_{t_k} に M_s を施して得られた個体が $M_s(a_{t_k}) = (\dots, m_{\chi_2}, \dots, m_{\chi_1}, \dots)$ のようになる。ただし、 χ_1, χ_2 は、 $\{1, 2, \dots, N\}$ に一様分布する整数乱数とする。

4. 計算機実験結果

協調型共進化 GA を以下の二つの DFLP に適用した。問題 I と問題 II はそれぞれ参考文献 [17] および参考文献 [6] から引用し、それぞれ Table 6 および Table 7 に示す。

- 問題 I : 5 生産期間, 2×3 の FLP
- 問題 II : 5 生産期間, 3×3 の FLP

突然変異率は個体に対して 0.2、集団サイズと世代数は評価回数が参考文献 [6] の実験と同じになるように設定している (評価回数 = 集団内個体数 $\times$ 集団サイズ $\times$ 世代数)。共進化する集団のサイズが探索結果に及ぼす影響を調べるため、同じ評価回数のもとで、異なる集団サイズを用いた。実験結果はすべて 100 回の平均結果とする。以下では、提案手法による実験結果を示す。

4.1 DFLP に関する実験結果

Table 1 には問題 I に関する結果を示している。この問題の最適解は 71,187 [17] である。実験 A に示されるように、提案手法はすべてのパラメータセットのもとで、最適解を発見することができた。

同じ問題に関して、参考文献 [17] では、DP を使って最適解より 1.1% 劣った解が見つかった。また、参考文献 [6] では、GA を使って同じ最適解を見つけている。

Table 2 には問題 II に関する結果を示している。実験 A は、再配置コストが考慮されない場合の結果を示している。再配置コストがない場合には、問題 II は五つの独立な SFLP に相当し、最適解として 592,029 [17] が知られている。Table 2 での実験 A に示すように、提案手法

Table 1 Results for Problem I. “I” and “P” in “I×P” stand for size of each subpopulation and the number of subpopulations, and “G” for the number of generations. “Best(Hits)” indicates the best objective function values found and the numbers of runs that the optimum has been found among 100 runs. “Mean Best” and “Std Dev” stand for the mean best objective function values of 100 runs and the standard deviation.

I×P	G	Best(Hits)	Mean Best(Std Dev)
5 × 5	800	71,187 (14)	71,762(389)
10 × 5	400	71,187 (4)	72,074(677)
20 × 5	200	71,187 (5)	72,417(930)
40 × 5	100	71,187 (9)	72,254(697)
80 × 5	50	71,187 (5)	72,404(776)

Table 2 Results for Problem II with no shifting cost. “I” and “P” in “I×P” stand for size of each subpopulation and the number of subpopulations, and “G” for the number of generations. “Best(Hits)” indicates the best objective function values found and the numbers of runs that the optimum has been found among 100 runs. “Mean Best” and “Std Dev” stand for the mean best objective function values of 100 runs and the standard deviation.

Exp.	I×P	G	Best(Hits)	Mean Best(Std Dev)
A	5 × 5	1600	592,277	595,295 (1,364)
	10 × 5	800	592,029 (2)	593,858 (1,304)
	20 × 5	400	592,029 (2)	593,918 (1,306)
	40 × 5	200	592,029 (2)	594,478 (1,464)
	80 × 5	100	592,775	595,635 (1,670)
B	5 × 5	2800	609,403	613,488 (1,702)
	10 × 5	1400	607,595	610,979 (1,496)
	20 × 5	700	606,774	611,254 (2,065)
	40 × 5	350	606,762	612,157 (1,775)
	80 × 5	175	608,857	613,342 (2,130)

は中間的な集団サイズ (10, 20, 40) を用いた時、最適解を見つけた。同じ問題に関しては、参考文献 [6] では GA を使って、1 回試行の結果として、最良解と平均解はそれぞれ 593,856 と 595,274 であり、最適解は見つからない。

提案手法による再配置コストを考慮した場合 DFLP の結果を Table 2 での実験 B に示している。得られた最良解のコストは 606,762 であり、Fig. 1 にその配置を示している。

参考文献 [6] は同じ問題に対して、各生産期間における再配置コストに制約を加えた場合、GA による最良解は 608,904 としている。Fig. 2 はその最良解の配置を示している。各生産期間における再配置コストに関する制

Period 1			Period 2			Period 3		
8	3	7	8	1	7	3	1	7
9	2	5	9	3	2	9	4	2
4	1	6	6	5	4	6	5	8

Period 4			Period 5		
8	1	9	4	6	9
4	3	2	5	3	2
6	5	7	1	8	7

Fig. 1 The best layout found in Exp. B with cost of 606,762. No fund constraint imposed.

Period 1			Period 2			Period 3		
7	3	8	7	1	8	8	1	3
1	2	9	2	3	9	2	4	9
4	5	6	4	5	6	7	5	6

Period 4			Period 5		
8	1	9	7	2	9
4	3	2	8	3	6
7	5	6	1	5	4

Fig. 2 The best layout reported in Conway *et al.*, (1994) [6] with cost of 608,904. With fund constraint.

約が実験結果に及ぼす影響を示すために、Table 4 には Fig. 1 と Fig. 2 に示した配置の各生産期間におけるコストを示している。Fig. 1 は提案手法による各生産期間における再配置コストに関する制約が課されない場合の最良解であるため、全コストは Fig. 2 に示した最良解より少ないが、第2, 4 生産期間での再配置コスト、および全再配置コストはそれより多い。

4.2 再配置コスト総額の制約付き DFLP に関する実験結果

提案手法を、(15) 式で定式化した再配置コスト総額の制約付き DFLP に適用した。参考文献 [6] での最良解の全再配置コスト 12,107 を再配置コスト総額に関する制約 B として用いた。 B を超える再配置に対し、ペナルティとしてその再配置コストの 100 倍のコストが課される。Table 3 には実験結果を示す。実験 D で得られた最良解を Fig. 3 に示す。また、各生産期間における運搬コストおよび再配置コストを Table 4 に示す。各生産期間における再配置コストに関する制約がないため、この最良解の第2, 4 生産期間の再配置コストは参考文献 [6] での最良解より多いが、運搬コストの合計と再配置コストの合計はそれより少ないことが分かる。

4.3 各生産期間に再配置コストの制約付き DFLP に関する実験結果

提案手法を、(16) 式で定式化した各生産期間に再配置コストの制約付き DFLP に適用した。各生産期間におけるそれぞれの再配置コストに関する制約は、参考文献 [6] には示されていないため、その最良解での再配置コスト (2,304; 2,349; 2,925; 4,539) を参考に、各生産期間に

Table 3 Results for Problem II with shifting cost. “I” and “P” in “I×P” stand for size of each subpopulation and the number of subpopulations, and “G” for the number of generations. “Best” indicates the best objective function values found among 100 runs. “Mean Best” and “Std Dev” stand for the mean best objective function values of 100 runs and the standard deviation.

I×P	G	Best	Mean Best (Std Dev)
5 × 5	2800	618,854	630,162 (4,872)
10 × 5	1400	608,768	620,662 (6,384)
20 × 5	700	608,610	622,938 (6,914)
40 × 5	350	607,427	626,461 (7,554)
80 × 5	175	611,261	633,535 (9,930)

Period 1			Period 2			Period 3		
7	4	6	7	2	4	7	2	8
3	2	5	1	3	5	1	4	5
8	9	1	8	9	6	3	9	6

Period 4			Period 5		
9	2	7	8	2	7
1	3	5	1	3	6
8	4	6	5	4	9

Fig. 3 The best layout found with found constraint 12,107

Period 1			Period 2			Period 3		
7	4	1	7	2	4	7	2	8
3	2	5	1	3	5	1	4	5
8	9	6	8	9	6	3	9	6

Period 4			Period 5		
9	2	7	8	2	7
1	3	5	1	3	6
8	4	6	5	4	9

Fig. 4 The best layout found with Period fund constraints (3,000; 3,000; 3,000; 3,000)

における再配置コストの制約として (3,000; 3,000; 3,000; 3,000) を用いた。各生産期間での再配置コストが B_t を超えた場合、ペナルティとしてその再配置コストの 100 倍のコストを課す。Table 5 には実験結果を示す。得られた最良解を Fig. 4 に、最良解の各生産期間でのコストを Table 4 にそれぞれ示す。ここから、コストの合計は参考文献 [6] での最良解より少ないことが分かる。

4.4 考察

以上の実験結果を通じて、提案手法は従来手法より、良い性能を示すことが分かった。これは、多集団による共進化で集団の初期収束を防ぎ、また、各集団内の個体が他集団内の個体と協調して生成した完全解によって評価され、一種の効率的な組換え操作と考えられるため、

Table 4 Cost comparison of the layouts in Fig. 1, Fig. 2, Fig. 3 and Fig. 4

Layouts	Period	1	2	3	4	5	Sum
Fig. 1	Material handling cost	116,295	115,795	120,977	120,110	120,770	593,947
	Shifting cost	-	4,450	1,581	3,272	3,512	12,815
Fig. 2	Material handling cost	118,875	115,795	122,353	119,004	120,770	596,797
	Shifting cost	-	2,304	2,349	2,925	4,529	12,107
Fig. 3	Material handling cost	117,043	115,795	120,977	120,110	121,802	595,727
	Shifting cost	-	3,714	1,581	3,272	3,133	11,700
Fig. 4	Material handling cost	118,203	115,795	120,977	120,110	121,802	596,887
	Shifting cost	-	2,804	1,581	3,272	3,133	10,790

Table 5 Results of Problem II with period fund constraints (3,000; 3,000; 3,000; 3,000). “I” and “P” in “I×P” stand for size of each subpopulation and the number of subpopulations, and “G” for the number of generations. “Best(Hits)” indicates the best objective function values found and the numbers of runs that the optimum has been found among 100 runs. “Mean Best” and “Std Dev” stand for the mean best objective function values of 100 runs and the standard deviation.

I×P	G	Best	Mean Best(Std Dev)
5 × 5	2800	619,041	635,001 (5,954)
10 × 5	1400	610,767	625,282 (7,199)
20 × 5	700	607,677	629,311 (8,889)
40 × 5	350	617,285	634,640 (10,170)
80 × 5	175	615,287	641,098 (11,540)

探索性が高められた。共進化する集団のサイズが探索結果に及ぼした影響は問題の設定によって異なるが中間的なサイズ(10,20,40)を用いたほうがよい。提案手法でのパラメータは、実際的に突然変異率だけとなるが、その値をすべての実験において固定しているため、提案手法がパラメータ依存性の少ない手法といえる。

なお、突然変異として、swap mutation operator が用いられたが、swap mutation operator は、同様に QAP によって定式化できる Traveling Salesman Problem [12] において、むしろ他の突然変異操作より性能が悪いことが報告されている [13]。本論文においては、共進化の効果を観察しやすくするために、簡単な遺伝操作を用いた。多くの GA による DFLP へのアプローチは組換え操作に重点をおいているが、提案した共進化 GA において、各集団内には、組換え操作は用いていない。しかし、DFLP 用に定義された組換え操作は簡単に導入することができる。より高度な遺伝操作を共進化 GA に導入することで、その問題解決能力のさらなる向上が期待できる。

5. おわりに

本論文では、動的施設配置問題 (Dynamic Facility

Table 6 Data for Problem I

Period 1	0	63	605	551	116	136
	63	0	635	941	50	191
	104	71	0	569	136	55
	65	193	622	0	77	90
	162	174	607	591	0	179
	156	13	667	611	175	0
Period 2	0	175	804	904	56	176
	63	0	743	936	45	177
	168	85	0	918	138	134
	51	94	962	0	173	39
	97	104	730	634	0	144
	95	115	983	597	24	0
Period 3	0	90	77	553	769	139
	168	0	114	653	525	185
	32	35	0	664	898	87
	27	166	42	0	960	179
	185	56	44	926	0	104
	72	128	173	634	687	0
Period 4	0	112	15	199	665	649
	153	0	116	173	912	671
	10	28	0	182	855	542
	29	69	15	0	552	751
	198	71	42	24	0	758
	62	109	170	90	973	0
Period 5	0	663	23	128	119	50
	820	0	5	98	141	66
	822	650	0	137	78	91
	826	570	149	0	93	151
	915	515	53	35	0	177
	614	729	178	10	99	0
Shifting cost	887	964	213	367	289	477

Layout Problems : DFLP) への協調型共進化 GA による一解法を提案した。協調型共進化 GA は Canonical GA と二つの点において異なる。まず、協調型共進化 GA では多集団を用い、集団内の個体は問題の部分解を表す。多集団による共進化によって集団の初期収束を防ぎ、探索性を高めることができる。個体の評価は他集団中の個体と協調的に生成した完全な問題解の良さによって行う。

Table 7 Data for Problem II

Period 1	0	3622	258	493	697	296	627	552	287
	991	0	316	443	570	684	334	283	1043
	673	6522	0	484	114	324	611	762	762
	791	4369	203	0	170	1031	598	923	788
	867	5146	56	203	0	1121	309	154	361
	894	3264	71	62	769	0	664	343	282
	714	3113	240	506	831	1183	0	1144	311
	588	1319	319	161	826	1194	744	0	773
	1096	6521	335	317	459	439	416	1222	0
Period 2	0	136	6371	886	1596	213	499	1378	476
	657	0	3461	1275	567	254	405	263	449
	590	528	0	488	498	273	311	1277	486
	179	684	1305	0	1748	101	462	1008	559
	772	550	6113	478	0	261	53	1134	1285
	511	822	2046	1105	1404	0	384	405	875
	577	690	2362	925	944	139	0	847	312
	300	461	3343	514	676	128	487	0	214
	291	560	6306	397	235	243	466	963	0
Period 3	0	265	720	3275	361	230	580	221	1433
	695	0	816	5276	636	683	637	1877	203
	901	1535	0	2322	323	592	129	857	979
	1138	298	987	0	400	1051	163	238	924
	619	478	856	4205	0	615	81	991	990
	647	1373	441	722	608	0	128	603	1040
	1008	1383	772	3552	497	836	0	1795	211
	1348	682	233	892	206	600	448	0	679
	1291	2281	595	3972	89	840	257	348	0
Period 4	0	753	632	1686	722	241	192	510	63
	840	0	897	795	3331	1274	426	611	442
	2138	895	0	1277	3019	693	88	470	514
	561	445	1444	0	1123	385	523	2015	428
	335	421	1549	560	0	820	251	1480	455
	636	515	776	1590	5257	0	781	504	416
	571	625	765	1304	5312	954	0	647	82
	1675	297	176	1137	1240	1313	715	0	321
	1187	1550	751	441	840	336	252	1695	0
Period 5	0	1017	663	1460	1118	804	256	1291	246
	854	0	1102	1476	1109	2931	975	1032	403
	850	1017	0	1503	412	4102	613	1083	140
	525	205	792	0	1060	3647	196	591	981
	1653	113	1133	1501	0	2160	203	706	695
	981	686	184	852	450	0	155	560	962
	781	1010	353	319	648	2043	0	914	185
	2031	701	930	755	1113	1883	772	0	175
	867	580	377	478	284	4879	106	325	0
	Shifting costs	802	985	517	500	736	910	768	923

つぎに、各集団内には Steady-State Reproduction が適用される。本論文では、従来研究との比較のため、二つの小規模の DFLP を用いて、提案手法の有効性を示したが、今後は、大規模の DFLP に提案手法の適用、また、組換え操作を採り入れる場合の提案手法の性能など

について検討したいと考えている。

参考文献

- [1] R. K. Ahuja, J. B. Orlin and A. Tiwari: A greedy genetic algorithm for the quadratic assignment prob-

- lem; *Computers & Operations Research*, Vol. 27, No. 10, pp. 917–934 (2000)
- [2] J. Bakakrishnan, F. R. Jacobs and M. A. Venkataraman: Solutions for the constrained dynamic facility layout problem; *E. J. Op. Res.*, Vol. 57, No. 2, pp. 280–286 (1995)
- [3] J. Balakrishnan and C. H. Cheng: Dynamic layout algorithms: a state-of-the-art survey; *Int. J. Mgmt. Sci.*, Vol. 26, No. 4, pp. 507–521 (1998)
- [4] J. Balakrishnan and C. H. Cheng: Genetic search and the dynamic layout problem; *Computer & Operation Research*, Vol. 27, No. 6, pp. 587–593 (2000)
- [5] M. M. Connors: Quadratic assignment problem algorithms and the location of indivisible facilities; *Management Sci.*, Vol. 13, No. 1, pp. 42–57 (1966)
- [6] D. G. Conway and M. A. Venkataraman: Genetic search and the dynamic facility layout problem; *Computers Ops. Res.*, Vol. 21, No. 8, pp. 955–960 (1994)
- [7] L. Davis: *Handbook of Genetic Algorithms*, Van Nostrand (1991)
- [8] K. A. De Jong and M. A. Potter: Evolving complex structures via cooperative coevolution; *Proc. 4th Annu. Conf. on Evolutionary Programming*, The MIT Press, pp. 307–317 (1995)
- [9] L. R. Foulds: Techniques for facilities layout: deciding which pairs of activities should be adjacent; *Management Sci.*, Vol. 9, No. 12, pp. 1414–1416 (1983)
- [10] F. S. Hillier: Quantitative tools for plant layout problem; *J. Industrial Engineering*, Vol. 14, No. 1, pp. 33–40 (1966)
- [11] T. C. Koopmans and M. Backman: Assignment problems and the location of economic activities; *Econometrica*, Vol. 25, pp. 53–76 (1957)
- [12] A. Kusiak and S. S. Keragu: The facility layout problem; *E. J. Op. Res.*, Vol. 29, pp. 229–251 (1987)
- [13] B. Manderick and P. Spiessens: The genetic algorithm and the structure of the fitness landscape; *Proc. 4th Int. Conf. on Genetic Algorithms*, Morgan Kauffman, pp. 143–150 (1991)
- [14] M. A. Potter and K. A. De Jong: A cooperative coevolutionary approach to function optimization; *Parallel Problem Solving From Nature - PPSN III*, *Int. Conf. on Evolutionary Computation*, Lecture Notes in Computer Science, Springer Verlag, Vol. 866, pp. 249–257 (1994)
- [15] M. A. Potter and K. A. De Jong: Evolving neural networks with collaborative species; *Proc. 1995 Summer Computer Simulation Conference*, Ottawa, Ontario, pp. 340–345 (1995)
- [16] M. A. Potter, K. A. De Jong and J. J. Grefenstette: A coevolutionary approach to learning sequential decision rules; *Genetic Algorithms: Proc. 6th Int. Conf.*, Morgan Kaufmann, pp. 366–372 (1995)
- [17] M. J. Rosenblatt: The dynamics of plant layout; *Management Sci.*, Vol. 32, No. 1, pp. 76–86 (1986)
- [18] J. A. Tompkins and J. A. White: *Facilities Planning*, Wiley (1984)