多目的遺伝アルゴリズムによる船体ブロック建造工程の最適化に関する研究

正員　荒井　誠*　　正員　長岡　知**

A Study on the Optimization of the Hull-block Construction Process by Multi-Objective Genetic Algorithm

by　Makoto Arai, *Member*　Tomo Nagaoka, *Member*

Summary

In modern shipyards, the block construction method is widely used in building ship hulls. But with this method, when work falls behind schedule the results include downtime, cost increases, and delivery delays. Therefore, an efficient schedule is strongly required. However, a large number of blocks with different sizes and shapes and different procedures must be considered. In addition, the existence of several objectives, such as total cost reduction, shorter completion period, efficient workforce allocation, make the problem more difficult. The block construction schedule is, in short, so complex that veteran engineers usually resort to a rule of thumb in preparing it. For these reasons, an obtained schedule is not necessarily an efficient one. To improve this situation, we must optimize schedules rationally and obtain practical solutions that satisfy all of these requirements simultaneously. Thus, the Multi-Objective Genetic Algorithm (MOGA) is utilized in this study. MOGA applies a Genetic Algorithm (GA) using the concept of Pareto optimization for multi-objective optimization. This method uses the GA characteristics of collective evolution of a solution group. This makes it possible to efficiently arrive at a practical solution from an almost infinite number of solution candidates. This study produced useful information, such as the trade-off relation between overtime work and a shortened completion period, the Pareto solution for total workforce and total completion period, and optimum worker assignment.

1．緒　　言

現在の我が国の造船所で主流となっている船体建造法は、ブロック工法と呼ばれるものである。ブロック工法では、船体を船体ブロック（以下ブロックと呼ぶ）と呼ばれる単位に分割して陸上の工場内で流れ作業により建造し、船渠・船台上において搭載・結合作業を行って船体を完成する。ブロック工場における作業は、鋼板の切断に始まり、部材の取付、溶接、先行艤装等の工程を経る。その際、ブロックを建造するスケジュールの適否は、船舶建造作業全体の効率を左右する重要な問題である。スケジュールが適切でない場合、ブロック工場内の各工程に作業量の繁閑が生じて効率が悪い上、ブロックの完成が遅れるため船舶そのものの竣工を遅延させかねない。このため効率的なスケジュールの作成が強く求められている。

しかしながらブロック建造スケジュールは、船舶の竣工引渡し日程、各ブロックの船渠・船台への搭載日程、ライン上各工程の設備能力、作業者数といった多数の事項による強い制約を受けている。また、最適化すべき項目として、工期やコストなど複数の項目が存在している。

*　横浜国立大学大学院
**　国土交通省
（研究当時　横浜国立大学大学院）

原稿受理　平成13年7月9日
秋季講演会において講演　平成13年11月15，16日

このため、ブロック建造スケジュールは、搭載日程を元に熟練計画者の勘と経験に頼って作成されているのが現状である。従って、効率の良いスケジュールを作成することが困難である上に、スケジュール作成には少なからぬ時間を要している。

ところで、近年、計算機を利用して造船所での作業の効率化や、作業者の支援を図る試みが種々行われている。ブロック建造スケジュールの作成に計算機を利用する場合、可能性のあるスケジュール候補の数が膨大であるため、全てのスケジュール候補について評価しようとするならば計算時間が無限に必要となり、事実上検討は不可能になる。しかしスケジュールの作成においては、真の最適解に十分近いと考えられる解を求めることが出来れば実用上は十分と考えられる。

そこで本研究では、多目的遺伝アルゴリズム（Multi-Objective Genetic Algorithm、以下、MOGA）を用いてブロック建造スケジュールを最適化することを試みた。MOGA は遺伝アルゴリズムをパレート最適の概念を用いて多目的最適化に応用したものであり、解集団の集団的進化という遺伝アルゴリズムの特質を利用しているため、効率的に最適解を得ることができる[1),2)]。研究の結果、残業量と工期短縮量との間のトレードオフ関係、作業者数と工期の関係を示すパレート解、最適な作業者配置等の有益な知見を得た。

2. 多目的遺伝アルゴリズム

現実の工学的問題においては、最適化すべき目的が1つではなく複数存在することが一般的である。こうした複数の目的を同時に最適化する問題は、多目的最適化問題と呼ばれている。本研究で扱うブロック建造のスケジュールを例に取るならば、工期の最短化とコストの最小化というように複数の目的を同時に考える必要がある。通常これらの要素は互いに相反する性質を持ち、最適化を行う場合にはそれぞれの要素の妥協を図ることが問題の本質となる。

多目的最適化問題は、一般に以下のように記述できる。

目的関数　$f_1(X),f_2(X),...f_q(X) \rightarrow$ 最大 or 最小

制約条件　$X \in S$

変数　　$X=(x_1,x_2,...x_n)$

S は、変数の存在範囲に対する制約である。

ここで、上記の問題の解法について検討してみる。まず、一つの方法として、q 個の目的関数に対して重み係数を $C_1,C_2,...,C_q$ と決めて、

$$T(X) = C_1f_1(X) + C_2f_2(X)+,...,+C_qf_q(X) \quad (1)$$

のように定義される T(X)を最大化または最小化するという単目的最適化問題への変換が考えられる。また、もう一つの方法として、目的関数の中で最も重要と思われる特定のもののみを最適化の対象とし、他の目的関数は一種の制約条件として取り扱い、一定の基準を満足させるだけにとどめるという方法も考えられる。

しかしながら、ただ一つの最適解しか得られない上記の二つの方法は、多目的最適化問題で最も重要な「複数の目的間でいかにトレードオフを考えるか」という点で十分なものとは言えない。また、重み係数をいかに決定するか、どの目的関数を最重要視すべきか等、実用上取り扱いが難しい問題がある。

以上の理由から、本研究では「パレート最適」という概念を用いて複数の目的関数間のトレードオフについて調べることができる多目的遺伝アルゴリズム（MOGA）を用いることとした。

2.1 パレート最適解

パレート最適解とは、ある目的関数の値を良くしようとすると少なくとも1つの他の目的関数の値が悪くなるような解である。

簡単のため2つの目的関数を最小化する問題を考える。Fig.1 において解 A は解 B に対しどちらの目的関数の点からいっても優れている。このような場合、解 A は解 B を支配している（解 B は解 A に支配されている）と言う。幾何学的には、ある解を原点としたときにその解の第3象限に他の解がなければ、それは他の解に支配されない解（パレート解）である。解候補が多数ある場合には、一般的にこのような他の解に支配されない解が複数個存在し、それらをパレート最適解群と呼ぶ。

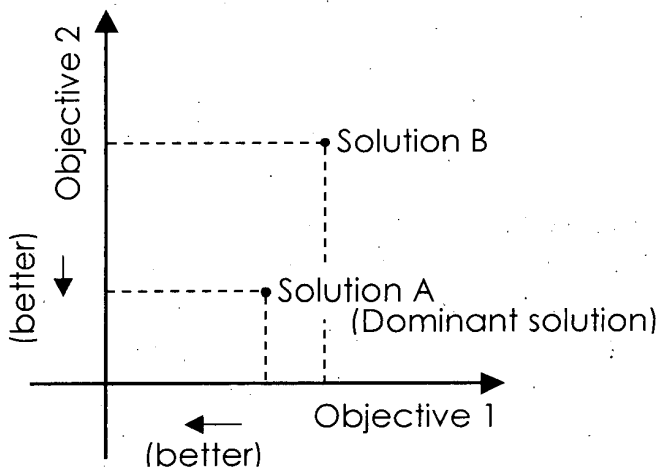


Fig.1 Relationship between two solutions

パレート最適解の集合であるパレート最適解群は、目的関数間のトレードオフの関係を示す。多目的最適化問題の本質は複数の目的関数間のトレードオフ関係を明

らかにすることにあるため、パレート最適解群を探索することにより多目的最適化の目的が達せられる[1),2),3)]。実際のシステム設計等に際してはパレート最適解の中からいずれかの解が選択される。これを選好解と呼ぶ。

2.2 パレート最適化概念の MOGA への適用

MOGA は、遺伝アルゴリズム(Genetic Algorithm, GA)が個体群を用いて解空間の探索を行うことを利用して、パレート最適解群を一括して求める方法である。

パレート最適化の概念を MOGA に導入するにあたっては「パレート・ランキング方式」が用いられる。すなわち、ある世代を構成する解の集団で、p_i 個の解により解 X_i が支配されるとき、解 X_i のパレートランク $rank(X_i)$ を

$$rank(X_i)=1+p_i \tag{2}$$

で定義する[1)](Fig.2 参照)。より「最適」に近い解ほどパレートランクは小さくなり、解集団内でパレート最適である個体のランクは 1 となる。GA による最適化計算における個体（解）の適応度 $T(X_i)$ を例えばパレートランクの逆数で与え、$T(X_i)$ を最大化（すなわちパレートランクを最小化）することにより多目的最適化を行うことができる。

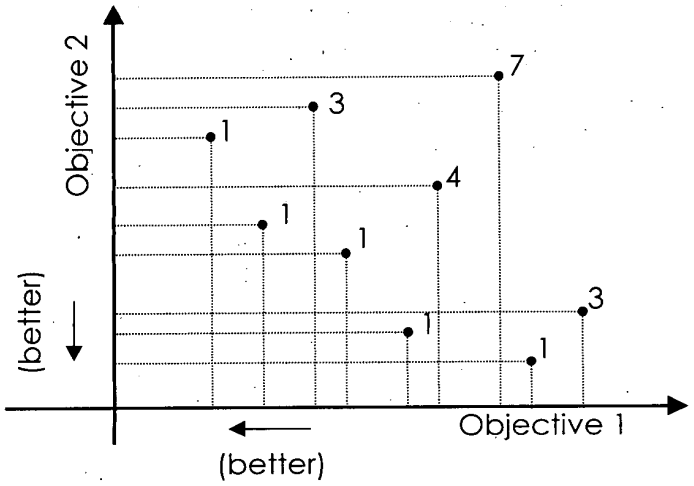


Fig.2 Multi-objective ranking

2.3 シェアリングについて

MOGA により最適化を行う場合には、計算の結果得られた解がパレート最適であることと同時に、トレードオフ関係を評価する立場からは、パレート最適解群が解空間においてできるだけ一様かつ広範囲に分布していることが望ましい。このため GA の計算過程において解の多様性を保つことが重要である。しかしながら、GA では乱数を用いて確率的に選択交配を行うため、同じパレートランクを持つ解のうちのある特定の個体のみが選ばれることにより解が解空間の一部に集中してしまい、解の多様性が失われる現象が起きやすい。ある程度最適化が進んだ段階で解の多様性が失われた場合、失われた多様性はなかなか回復しない。これを避けるため、解空間上の多くの解が集中している部分で解の適応度を意図的に下げるシェアリング[1),2),3),4),5)]と呼ばれる操作が必要となる。即ち GA の確率的選択により解空間の一部に「集まって」しまった解を「散らして」やることである。

解 X_i に対する適応度 $T(X_i)$ にシェアリングを施して修正した適応度 $T'(X_i)$ は、個体数を N として、

$$T'(X_i)=\frac{T(X_i)}{\sum_{j=1}^{N} s(d(X_i,X_j))} \tag{3}$$

と定義できる。ここで s は解がどの程度近くに集中しているかを示す関数であり、シェアリング関数と呼ばれる。これまでに種々のシェアリング関数が考えられており、例えば解空間上で、ある距離より近くにある解の数に重み付けを行ったものとして次式のようなものがある[2),5),6)]。

$$s(d)=\begin{cases}1-\left(\dfrac{d}{\sigma_{share}}\right)^{\alpha} & d<\sigma_{share}\\ 0 & d\geq\sigma_{share}\end{cases} \tag{4}$$

ここで、$d=d(X_i,X_j)$ は解空間における X_i,X_j 間の距離をあらわす。また、σ_{share} はニッチ数[1),2),3)]と呼ばれる定数であり、解が互いにどの程度近くにあるときに評価を下げるかを決定するしきい値である。σ_{share} の決定法として各種の提案がなされているが、本研究では各目的関数の個別の最大値 M_k と最小値 m_k を使用し、以下の式により求める[1)]。

$$N\sigma_{share}^{q-1}-\frac{\prod_{k=1}^{q}(M_k-m_k+\sigma_{share})-\prod_{k=1}^{q}(M_k-m_k)}{\sigma_{share}}=0 \tag{5}$$

ただし N は個体数である。また、べき乗数 α は最適化の結果からトライアンドエラーにより決定することとした。

3．ブロック建造スケジュールの最適化

3.1 最適化プログラムの概要

本研究では、ブロック建造作業を模擬しスケジュールを最適化するプログラムを開発した。開発したプログラムはスケジュールを元に建造作業のシミュレーションを行う「シミュレーション部分」と、MOGA を用いて

スケジュールの最適化を行う「最適化部分」とに分けられる。両者の関係はFig.3に示す通りである。以下、それぞれについて説明する。

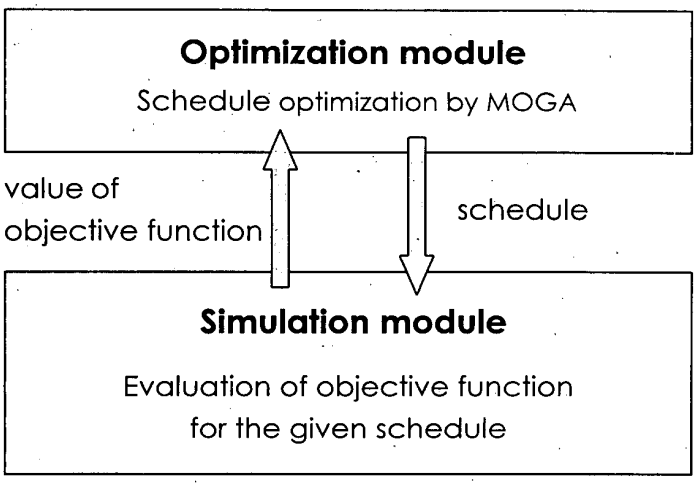


Fig.3 Schedule simulation program by MOGA

3. 1. 1 シミュレーション部分について

シミュレーション部分は最適化部分から受け取った遺伝子配列（スケジュールを示している）に従ってブロック建造作業のシミュレーションを行い、所要の目的関数の値を計算して最適化部分に渡す。

実際の造船所のブロック建造工場には数多くの制約条件が存在し、全ての条件を考慮してシミュレーションを行うことは不可能である。そのため、本研究では条件を単純化した工場モデルを用いてシミュレーションを行う。工場モデルの作成にあたっては実際の造船所の平板ブロック工場を参考にした。モデル化したブロック建造ラインをFig.4に示す。建造ラインには工程1〜6の6つの工程がある。工程（1,2,3,4,5,6）はそれぞれ、（板取付工程，板溶接工程，枠取付工程，枠溶接工程，枠溶接工程，艤装工程）と仮定している。

各ブロックの建造作業は工程1から始まり、工程6での作業を終了するとそのブロックは完成したものとする。枠溶接の工程は工数を要するため工程4と工程5を平列に置いており、各ブロックにはどちらか一方で枠溶接の作業が行われる。すなわち各ブロックは工程（1,2,3,4,6）または工程（1,2,3,5,6）のいずれかの経路を通って完成する。それぞれを経路1、経路2と呼ぶことにする。なお、プログラムの仕様上は、ブロック特性を考慮してそれぞれのブロックに対し経路を指定することもできるが、実際の計算では全てのブロックはいずれの経路も取ることができ、とるべき経路はスケジュールにより与えられるとした。

プログラム上、この建造ラインではブロックを以下のように処理することとした。

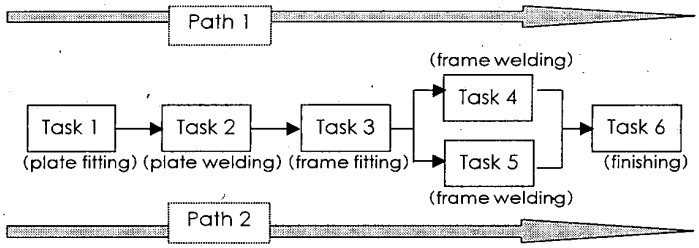


Fig.4 Model of block construction line

① 各ブロックには各工程に要する工数が与えられている。

② 各工程上には与えられた数の作業者が存在する。1人の作業者は、自分がいる工程に作業中のブロックがある限りは常に1時間に1「工数」の作業を行う。

③ 各ブロックには、進むべき経路がスケジュールで与えられている。

④ あるブロックはある工程での作業が開始されてから（当該ブロックの当該工程での所要工数 / 当該工程上の作業者数）だけ時間が経過すると、当該工程での作業を完了する。

⑤ ある工程での作業が完了したブロックは、最適化部分で決定されたスケジュールにより定められている経路に従って次工程に進む。ただし次工程上にブロックが存在する場合には進むことは出来ず、現工程上で待機する。待機中のブロックは次工程が空きしだい次工程に進み、作業が開始される（Fig.5参照）。

⑥ 第6工程での作業を完了したブロックは竣工したものとし、第6工程での作業完了時刻をブロックの竣工時刻として記録する。

⑦ ある時刻において第1工程上にブロックが存在しない場合、最適化部により作成されたブロック建造順序が示す次のブロックの建造が第1工程にて開始される。

⑧ 以上を、竣工したブロック数が建造予定ブロック数と等しくなるまで繰り返す。

計算に使用した船体ブロックのデータは、参考とした工場のラインで実際に建造された船体ブロック103個のデータをそのまま使用した（Table1）。また、各工程上の作業者数もこれらのブロックが製作された時期の作業者配置を参考にして決定した(Table2)。

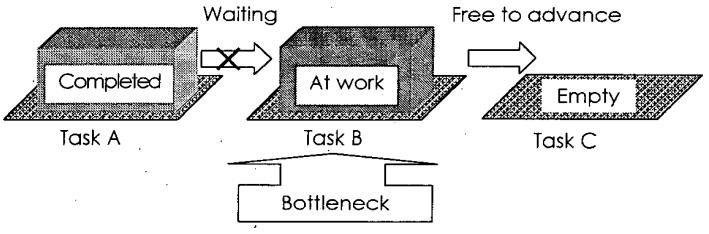


Fig.5 Construction process

Table 1 Data of construction blocks

Block No.	Ship No.	Man-hours				
		Plate fitting	Plate welding	Frame fitting	Frame welding	Finishing
B1	Ship 1	3.0	14.0	27.0	153.0	11.0
B2	Ship 2	2.0	9.0	14.0	67.0	8.0
B3	Ship 3	2.0	9.0	22.0	95.0	12.0
B4	Ship 2	2.0	12.0	15.0	67.0	8.0
B5	Ship 3	2.0	9.0	22.0	95.0	12.0
B6	Ship 3	5.0	22.0	24.0	101.0	11.0
B7	Ship 2	2.0	15.0	18.0	88.0	7.0
B8	Ship 3	4.0	22.0	24.0	84.0	10.0
B9	Ship 2	2.0	13.0	27.0	88.0	6.0
B10	Ship 3	4.0	20.0	20.0	95.0	10.0
:	:	:	:	:	:	:
B99	Ship 3	4.0	12.0	16.0	45.0	8.0
B100	Ship 2	2.0	13.0	13.0	75.0	9.0
B101	Ship 2	2.0	13.0	16.0	90.0	7.0
B102	Ship 2	2.0	13.0	14.0	76.0	9.0
B103	Ship 2	3.0	14.0	17.0	108.0	7.0

Table 2 Worker allocation of the construction line

Task No.	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Number of workers	2	3	4	7	7	2

3.2 最適化部分について

最適化部分はスケジュールをシミュレーション部分に与え、シミュレーション結果を元に MOGA によりスケジュールの最適化を行う。

最適化部分ではスケジュールを遺伝アルゴリズムの遺伝子として取り扱う。おのおのの遺伝子は、

① ブロックの建造順序を示す配列

② 各ブロックの通るべき工程経路を示す配列

の 2 つの配列からなる。例えばブロックの数がブロック 1～ブロック 8 までの 8 つ、各ブロックの通るべき経路の数が経路 1、経路 2 の 2 つである場合、遺伝子は

① (1,4,6,3,2,8,7,5)

② (1,2,2,1,1,1,1,2)

のような 2 つの配列の組である。

最適化部分は遺伝子（スケジュール） X_i をシミュレーション部分に渡し、シミュレーション結果に基づく q 個の目的関数の値

$$F(X_i)=\{f_1(X_i),f_2(X_i),\ldots,f_q(X_i)\} \quad (6)$$

から、各遺伝子のパレートランク rank(X_i)を算出し、評価関数 T(X_i)を

$$T(X_i)=\left\{\frac{1}{rank(X_i)}\right\}^{\beta} \quad (7)$$

として計算する。β はスケーリング係数と呼ばれる定数で、本研究では試行錯誤により 2.0 と定めた。

シェアリングは先に示した(3)、(4)および(5)式により評価関数に修正を施すことで行った。(4)式の係数 α は計算の結果を元に 1.0 とした。解間の距離についてのしきい値 σ_{share} の値は、初期解における各目的関数の個別の最大値 M_k と最小値 m_k を用いて(5)式により求めた。

最適化部分は(3)式の評価関数 T'(X)を最大化する X_i を探索する。探索手法はほぼ単純 GA[7]そのままである。淘汰処理には選択確率が T'(X_i)の値に比例するルーレットルールを用いた。交叉処理ではブロックの建造順序を示す配列については部分一致交叉（PMX、Partially Matched Crossover）法[8]を、ブロックの通るべき工程経路を示す配列については 2 点交叉を用いた。突然変異は配列の要素のうち乱数で選択された部分を切り出して向きを逆にし、元の位置に戻すことによって行っている。

3.3 解空間の規模

本問題の解になる可能性のある組み合わせ（スケジュール）は、103 個のブロックの建造開始順序を示す配列と 103 個のブロックがそれぞれ 2 本あるうちのいずれの経路を通るかを示す配列により表現されるので、その数は、

$$103!\times 2^{103}=1.00\times 10^{195}$$

という膨大なものとなる。本研究において計算に使用した計算機（CPU：Celeron700MHz）では本問題の目的関数の計算を毎秒約 102 回行うことが出来る。この計算機を用いて 1.00×10^{195} 個の解をしらみつぶしに調べる場合の計算時間は、

$$0.98\times 10^{193}\text{（秒）}=3.12\times 10^{185}\text{（年）}$$

となり、全ての解をしらみつぶしに調べることは事実上不可能である。

4．最適化計算結果

上記の最適化プログラムを用いて、3 通りのシミュレーションを行った結果を以下に示す。

4.1 残業を考慮した場合

残業を行った場合の工期短縮量と残業量との関係を調べるため、ブロック建造工期と残業工数を目的関数として最適化シミュレーションを行った。

残業を与えるアルゴリズムは、

① 1 日の労働時間を 8 時間とする。

② 1 日の終わりに作業中であるブロックのうち、ボトルネックになっている工程を調査する。ここで、ボトルネックになっている工程とは、前工程はブロッ

クの作業が終了していて待機状態にあり、次工程にはブロックが存在しない工程の事であるとする（Fig.5）。

③ ボトルネックになっている工程のうち、現在行っている作業が終了するまでの時間が最も短い工程1つで残業を行い、当日中に作業を終了させる。

④ ただし、1日あたり最大の残業時間は4時間以下とする。

というものである。

第1の目的関数であるブロック建造工期は、最初のブロックの起工から最後のブロックの竣工までに要した時間である。ただし残業を行った時間は含まない。

第2の目的関数としての残業工数は

残業工数=Σ(残業した人数×残業した時間)

と定義した。

作成した最適化プログラムにより、この2つの目的関数を最適化した。一世代当たりの個体数は8192、世代数は40世代である。計算は先に述べた性能の計算機を用いて約56分を要した。

計算により得られたブロック建造工期と残業工数との間の関係を示すグラフをFig.6に示す。横軸がブロック建造工期（時間）、縦軸は残業工数である。ただし、この例では、2章で述べたシェアリングの効果を検証するため、シェアリングなし（Fig.6.(a)）とシェアリングあり（Fig.6. (b)）の両方の結果を示している。

図より明らかなように、シェアリングありの結果の方がより最適な部分に解が広く分布している。また得られたパレート解もより良好なものとなっている。この原因としては、局所最適解近傍に集中していた解がシェアリング操作により局所解を脱したためと推察される。このことから、本問題のようなスケジューリング最適化問題へのシェアリングの適用は、解の多様性を保ち、より良い解を得るために有効な手段であると言える。従って、本論文の以下のシミュレーション例においては全てシェアリングを行った結果を示すことにする。

Fig.6(b)からは、残業工数を増やすとブロック建造工期が短くなる、という明らかな関係が読みとれる。パレート解群の分布がほぼ直線状であることから、残業量と工期短縮量との関係を予測することが可能である。また、残業工数の増加で意味があるのは200工数程度までであり、それ以上の残業はあまり工期の短縮に寄与しないということも分かる。

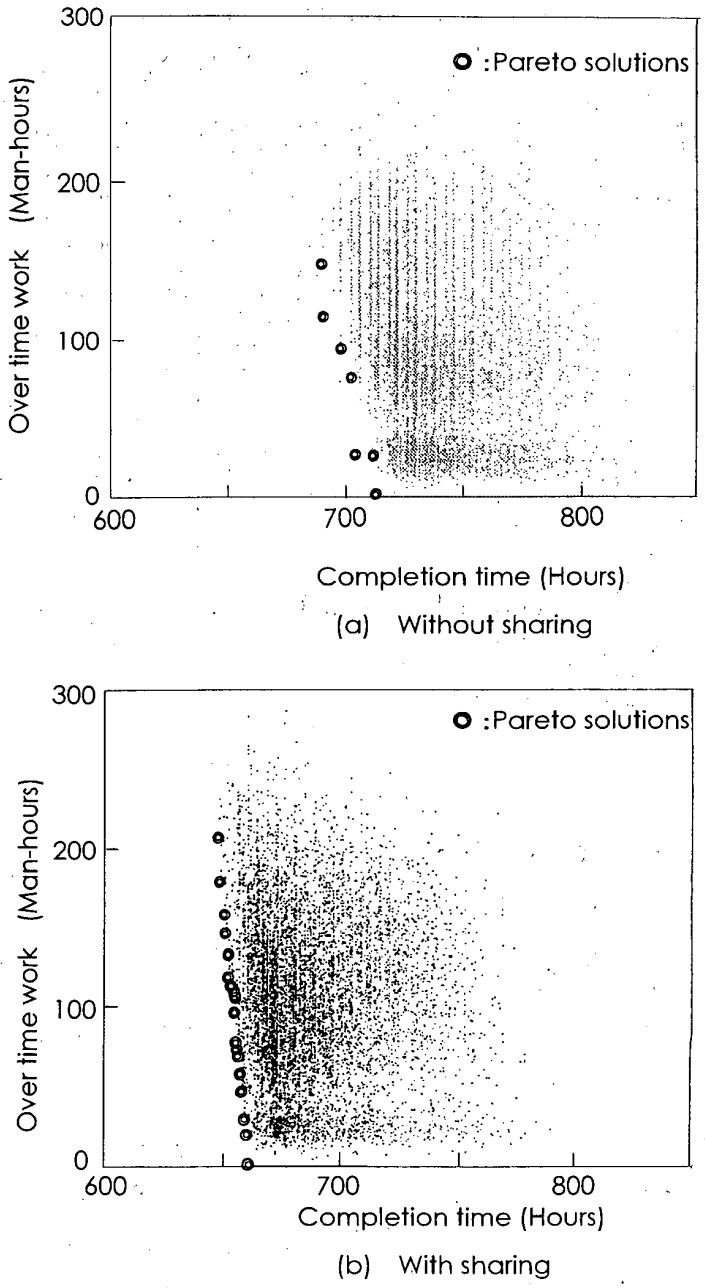


Fig.6 Relationship between completion time and overtime work

4.2 船台・船渠へのブロック搭載日程を考えた場合

次の例では、ブロック建造の総工期と個々のブロックの搭載日程超過との関係を調べた。

第1の目的関数としたブロック建造工期は、最初のブロックを起工してから最後のブロックが竣工するまでに要した時間として定義する。単位は時間である。

第2の目的関数とした搭載超過ペナルティとは、あらかじめそれぞれのブロックjに船台・船渠への搭載日程（=このブロックの完成期限）DDAY(j)が与えられているとして、シミュレーションにより求められたブロックの竣工時刻 DFIN(i,j)が完成期限を超過している場合にペナルティを科すと考える。本研究ではNBをブロック

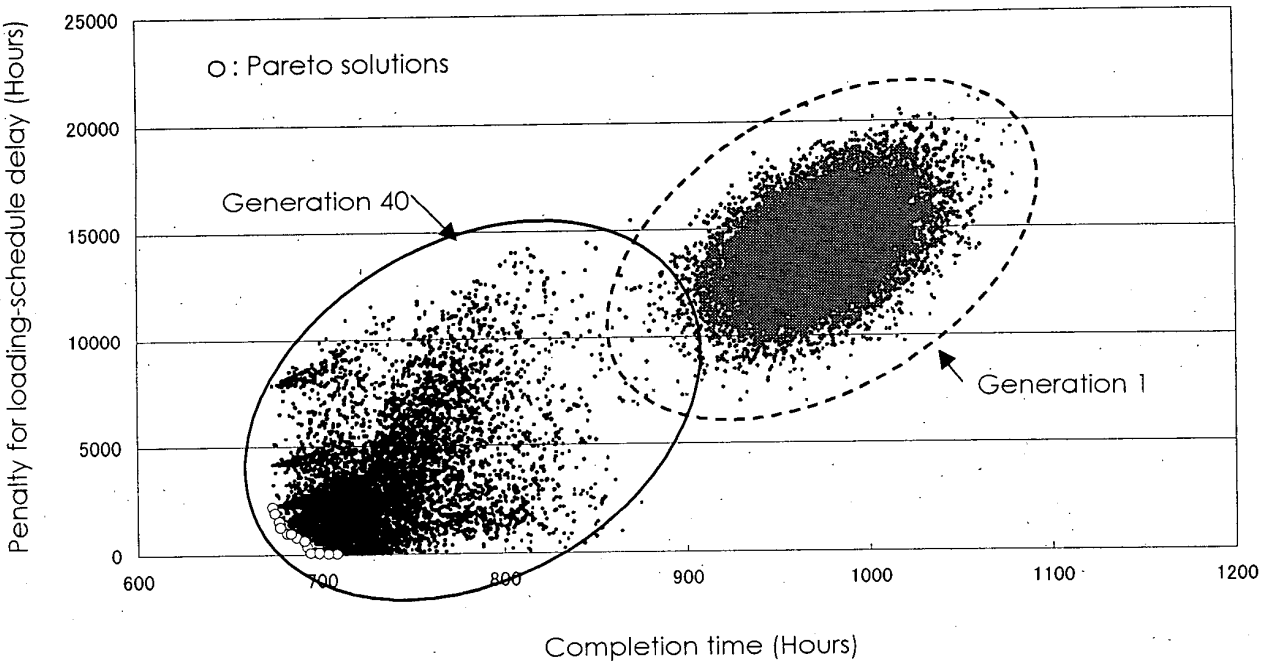


Fig.7 Relationship between completion time and penalty for loading-schedule delay

数として、

$$P(x_i) = \sum_{j=1}^{NB} DLAT(i,j) \tag{8}$$

$$DLAT(i,j) = \begin{cases} DFIN(j) - DDAY(i,j) & DFIN(j) > DDAY(i,j) \\ 0 & DFIN(j) \le DDAY(i,j) \end{cases} \tag{9}$$

により搭載日程超過ペナルティ $P(X_i)$を計算することとした。$P(X_i)$の単位は時間である。

作成した最適化プログラムによりこの2つの目的関数を最小化した結果をFig.7に示す。個体数は8192、世代数は40世代であり、計算に要した時間は約56分であった。同図には、初期解（第1世代）と最適化計算終了時の第40世代の解のそれぞれ8192組のスケジュール評価値がプロットされている。第40世代解の左下にある○印はパレート解群を示す。これらのパレート解は、例えば、あるブロックを当初予定されていた搭載日程より遅れて搭載するようにスケジュールを変更することによって、全体としての建造工期短縮を達成できることを示している。すなわち、個々のパレート解を現場のスケジュール担当者が吟味することにより、総建造工期短縮の可能性を検討することができよう。

なお、この例ではパレート解の存在する範囲はかなり狭い。この原因としては、Fig.7の初期解の分布を見ると分かるように、ブロック建造工期と搭載日程超過ペナルティの間に比例に近い関係があることが考えられる。すなわち、本問題の2つの目的関数間には、ブロック建造総工期を短縮するようなスケジュールを組むと自然に搭載日程に間に合うブロックが増えて搭載日程超過ペナルティが減少するという傾向がある。このため、解は右上がりの方向に存在範囲をもち、パレート解も比較的狭い範囲にしか存在しないと考えられる。

4.3 工程上の人員の増減を考えた場合

前節までの結果では各工程への作業員の配分に仕事量との不均衡があり、アイドル発生の原因の一つとなっていた。そこで各工程への作業者数の最適配置も含めた検討を行った。

この例では、これまでの2つの配列に加えて各工程の作業者数を表す配列を遺伝子に追加した。ただし各工程の作業者数はそれぞれ1~20人の間の値とした。

第1の目的関数であるブロック建造工期は最初のブロックの起工から最後のブロックの竣工までに要した時間である。

第2の目的関数として、全工程の合計の作業者数（総作業者数）を考える。

最適化プログラムによりこの2つの目的関数を最小化した。個体数は8192、世代数は50世代であり、計算には約70分を要した。

計算により得られた最終結果を2つの目的関数に対してプロットしたものをFig.8に示す。横軸はブロック建造工期（時間）、縦軸は総作業者数（人）である。

総作業者数6人から84人までの間で総作業者数とブロック建造工期に対するパレート最適解が得られた。作業者数の増加によるブロック建造工期の短縮効果は作業者数が少ないほど顕著であり、総作業者数が多くなる

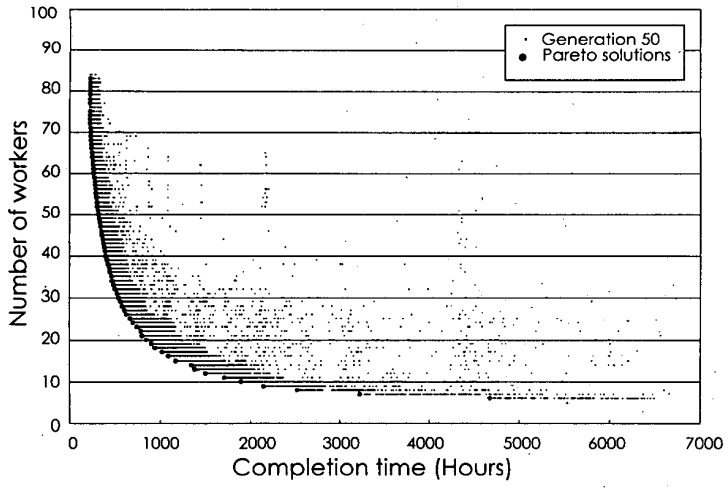


Fig.8 Relationship between completion time and total number of workers

と作業者数増加の建造工期短縮への寄与が弱まることが見て取れる。

総作業者数 6 人、12 人、18 人、30 人、54 人のそれぞれのパレート解における各工程への作業者数配置を Fig.9 に示す。各工程への最適な人数配分は、総作業者数が異なってもほぼ同様の比率であることがわかる。

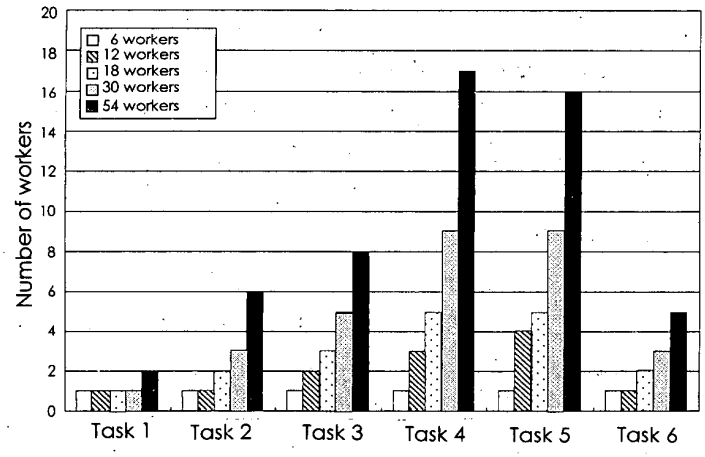


Fig.9 Optimum allocations for 6 to 54 workers

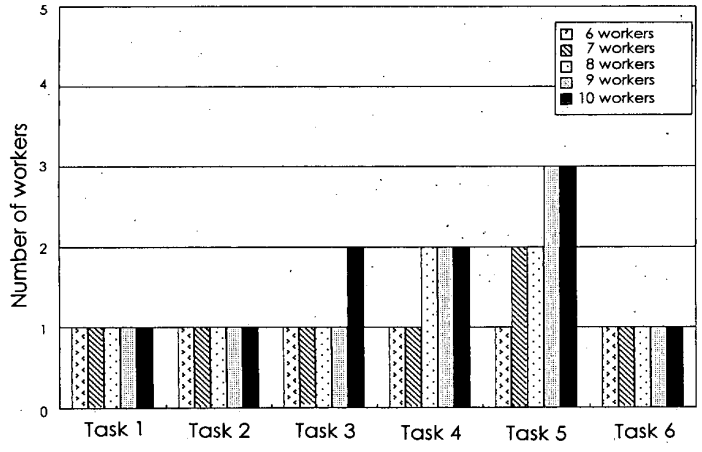
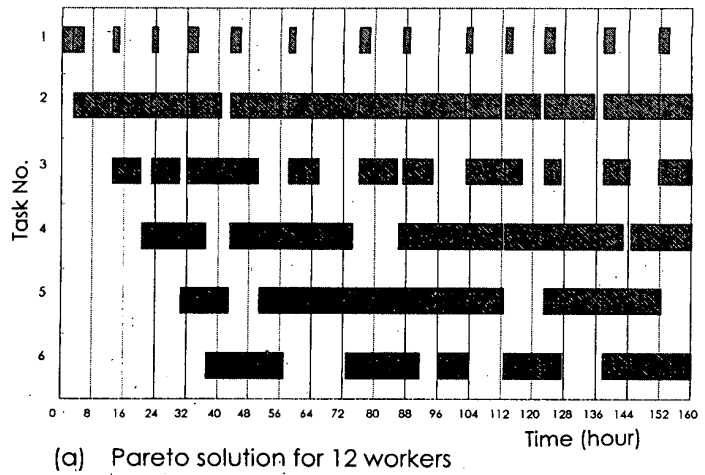
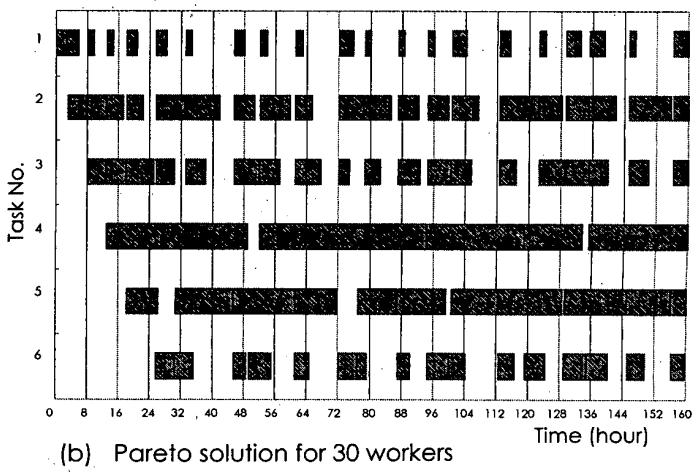


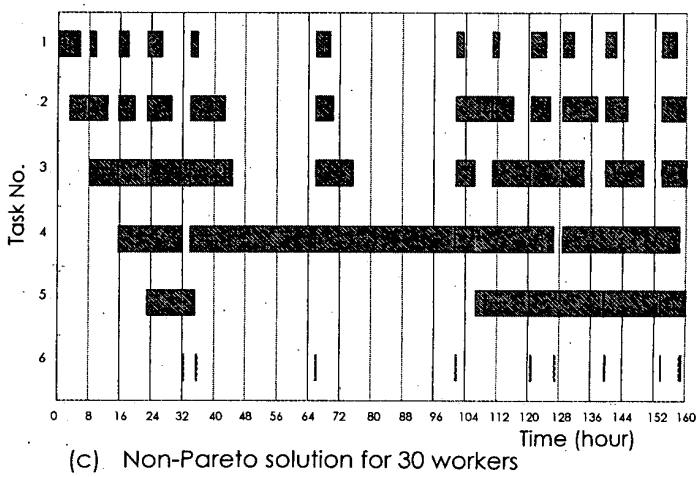
Fig.10 Optimum allocations for 6 to 10 workers



(a) Pareto solution for 12 workers



(b) Pareto solution for 30 workers



(c) Non-Pareto solution for 30 workers

Fig.11 Examples of block flows

次に、総作業者数 6 人から 10 人のパレート解での各工程への作業者の配置を示すグラフを Fig.10 に示す。この図から、作業者を 1 人増やすならばどの工程に増員するのが最適であるかが分かる。

また、総作業者数 12 人と 30 人のパレート解について、最初のブロックの起工から 160 時間経過までの作業の進行状況を Fig.11(a)および(b)に示す。横軸は時間、縦軸は工程を示す。図の塗りつぶし部分が各工程において作業実施中の時間帯を表し、塗りつぶしてない部分がア

イドルとなっている時間帯を表している。Fig.11(b)より、総作業者数の増加によって Fig.11(a)に比べアイドルとなる時間が減少することがわかる。また、このラインでは工程 2,4,5 がボトルネックになりやすく、そのため工程 1,6 にアイドルが生じやすいことが見て取れる。

さらに、総作業者数 30 人にもかかわらずブロック建造工期が総作業者数 12 人並みの効率の悪いスケジュール（非パレート解）を選び出し、最初のブロックの起工から 160 時間経過までの作業進行状況を Fig.11(c)に示す。この解では各工程への人員配置が（工程 1,工程 2,工程 3,工程 4,工程 5,工程 6）=（1,2,2,3,4,18）と非常にバランスが悪いこと、工程 4 にばかりブロックを流していること、ブロック建造開始順序が悪いこと等から、非常に多くのアイドルが発生している。

5. 結　言

本研究では、造船所におけるブロック建造スケジュールを多目的遺伝アルゴリズム（MOGA）により最適化することを試み、以下の結果を得た。

(1) 造船所のブロック建造スケジュール最適化に適したシミュレーションプログラムを開発し、本問題への MOGA の適用が有効であることを確認した。

(2) 複数の目的に対する最適化においては、問題により解空間の形状が大きく異なり、目的関数どうしの性質が類似するものと相反するものでは、パレート解の傾向も異なることが分かった。

(3) 残業を行った場合の残業量増減による工期増減量を示すパレート解を得た。

(4) 各工程の作業者数最適配置および作業者総数と工期との関係を得た。

(5) MOGA による多目的最適化の目的であるパレート解を求めるにあたり、シェアリングが有効であることを確認した。

謝　辞

本研究に対して、石川島播磨重工業（株）の太田垣由夫氏、柿元督孝氏から多数の有益な御討論を頂きました。厚くお礼申し上げます。

参 考 文 献

1) C.M.Fonseca and P.J.Fleming : Genetic algorithms for multiobjective optimization: Formulation, Discussion and Generalization", Proc. of 5th Int. Conf. on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, pp.416-423 (1993).

2) 大林茂:多目的遺伝的アルゴリズムによる空力最適設計、試験水槽委員会シンポジウム「船型設計と流力最適化問題」、 日本造船学会、pp.111-140 (1999).

3) 電気学会 GA 等組合せ最適化手法応用調査専門委員会：遺伝アルゴリズムとニューラルネット（スケジューリングと組合わせ最適化）、コロナ社(1998).

4) J.Horn, N.Nafpliotis and D.E.Goldberg : A niched Pareto genetic algorithm for multiobjective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, pp.82-87 (1994).

5) 廣安知之、三木光範、渡邉真也、畠中一幸：多目的分散遺伝的アルゴリズムにおけるシェアリング、収束判定、及び解の評価手法の検討、同志社大学理工学研究報告、Vol.40, No.4, pp.19-30 (2000).

6) Goldberg, D.E. : Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley Publishing Company, Inc. (1989).

7) L.デービス：遺伝アルゴリズムハンドブック、森北出版(1995).

8) 北野宏明：遺伝的アルゴリズム、 産業図書(1995).