

ジグ・工具の交換を考慮した JSP の LCO による解法

旭川工業高等専門学校 ○松村 有祐, 渡辺 美知子, 古川 正志

これ迄に TSP 及び JSP に対して LCO を適用し, その有効性を検証してきた. 数値計算実験により, LCO は GA よりも高速かつ高精度なスケジューリング解導出アルゴリズムであることが実証された. しかし, 現実的なスケジューリングシステムとして運用するためには, ジグ・工具割当の考慮は不可欠である. 本研究では 3 次割当問題となる, ジグ・工具の交換を考慮した JSP に対して, LCO を適用する方法を提案し, その有効性を数値計算実験で検証する.

1. 諸言

ジョブショップスケジューリング問題(Job-Shop Scheduling Problem; JSP)は, NP 完全問題として知られる組み合わせ最適化問題の一つである.

これまで, 著者らは遺伝的アルゴリズム(Genetic Algorithms, GA)や免疫アルゴリズム(Immune Algorithms, IA)を用いた JSP の解法を開発してきた. GA 及び IA は従来のディスパッチングルールによる方法と比較し, 高精度な近似解を導くことが実証されているが, GA よりも高速かつ高精度な近似解を導くアルゴリズムは未だ開発されていない. また, ジグ・工具の割当を考慮したスケジューリングを行う場合は, 問題の記述が複雑になるため, 良好なスケジューリング解を得るためには, GA でも多くの計算時間を要している.

本研究では, 巡回セールスマン問題(Traveling Salesman Problem, TSP)及び JSP などの一般的な組合せ最適化問題に適用可能な解法である局所クラスタリング組織化法(Local Clustering Organization, LCO)を提案し, JSP に適用する. ついで, LCO の演算において有効なジグ・工具の割当方法を示す. GA と LCO の性能を数値計算実験で比較し, ジグ・工具の割当を考慮した JSP における LCO の優位性を示す.

2. ジョブショップスケジューリング問題(JSP)

JSP は, n 個の仕事, m 台の機械を持つ工場で, 各機械に各仕事を割り当てる順序を決定し, 最適な工場スケジュールを求める問題である. 各仕事には機械の使用順序や納期が与えられるため, TSP に比べ制約条件が多い. ここでは, 仕事数 n , 機械数 m の JSP を n/m 問題と呼ぶ. ジグ・工具の割当を考慮する場合は, ジョブ・機械の順序決定問題だけでなく, ジグ・工具を効率的に機械に割り当てる必要があり, 問題の記述は更に複雑になる.

2.1. 従来の解法

JSP の解法には, 厳密解法では分岐限界法(Branch and Bound Method, BBM)¹⁾, 近似解法ではヒューリスティックな方法としてディスパッチングルール²⁾, 進化計算として GA の適用法が提案されている.

BBM は与えられた問題を子問題に分割し, その子問題の解が親問題の下界及び上界を導く簡単な問題とするもので, 子問題は同様に次々と新たな子問題へと分割される. 分割ができない子問題の下界が他のすべての子問題の下界よりも優れる場合に計算を終了する. ただし BBM は, 大規模問題において, 下界の分割数が飛躍的に増大し, 時間制約やマシンリソースの面から計算が不可能になる場合が多い.

ディスパッチングルールは, ある決められた規則に基づき, 仕事の処理順序を逐次決定し, 最終的にスケジュール解を得る. 例えば, ルールの一つである SPT は, 加工時間最小の仕事から順にスケジュールを構築する. 他にも MWKR や LWKR など多くのルールが存在する. しかし, いずれも解精度の面で, 近年研究されてきた GA などの進化計算に劣るが, 計算時間は高速である利点を持つ.

GA 及び IA などの進化計算は JSP に対してこれまで最も有効な解法として研究されてきた. GA は JSP のある解を一個体とみなし, 遺伝子としてコーディングし, 突然変異や交叉などの遺伝演算を与えながら, 各個体間で競争させる. あらかじめ定めた終了条件に達したときに, 最も優れた個体を近似解として得る. コーディング方法の一例として多重遺伝子がある. 一般的な n/m 問題を多重遺伝子でコーディングした場合, その遺伝子長は $n*m$ となる.

IA は GA により得られた解を免疫として記憶し, 生物界における免疫の作用を模倣し, GA が発生する類似解を抑制しつつ解を進化させる. 多峰性問題である JSP に対しては, GA よりも有効な方法とされているが, 計算時間は GA より必要となる.

これらから, JSP は組合せ数が爆発を解決するため, ルールベースによるスケジュール方法よりも, 進化・学習的に解を得る多点探索でない方法が望まれる.

3. 局所クラスタリング組織化法(LCO)

3.1. LCO の基底理念とアルゴリズム

著者らは TSP の新解法として LCO を開発した. LCO のアルゴリズムは, 局所的改善の繰り返しにより大域的改善を実現するリカッチ型学習方程式³⁾に基づく. 以下に n 都市の TSP に対する LCO の基本的アルゴリズムを示す.

$\Pi(i, j)$ を都市対 (i, j) 間の開経路とする. $P_i (i=1, 2, \dots, n)$ を連結都市群の先頭より i 番目の都市とする. 近傍係数を学習回数 t の関数として $d(t) (d(t) < n/2)$ とする. ただし, $d(t)$ は定数としても問題ない.

- (1) n 個の都市をランダムに一周経路として生成する.
- (2) ランダムに P_c を決定する. $\Pi(c-d(t), c+d(t))$ をもつ.
- (3) $\Pi(c-d(t), c+d(t))$ をクラスタリングにより最適化する.
- (4) 必要ならば, $d(t)$ を変化させ, (2)へ戻る. ただし, (3)の操作で解が改善しない場合が連続で発生し, それが指定回数に達した場合は計算を終了する.

ここで, (3)におけるクラスタリングの方法が明らかではないが, TSP に有効な方法として, 交換法(Simple Exchange Method, SEM), 逆位交換法(Inverse Exchange Method, IEM)及び平滑法(Smoothing Method, SM)を開発した.

LCO は TSP に対し, 近似解の精度及び計算速度の面から, 高い有用性を持つことをすでに実証している. JSP も TSP と同様に, 1つの解は1つの遺伝子として一意に表現される(多重遺伝子表現)⁴⁾ため, 遺伝子のクラスタリングがベースである LCO は JSP に対しても適用可能である.

3.2. JSP への適用

LCO を JSP に適用するために, まずスケジュール解を解表現, 即ち遺伝子として表現する必要がある. 本研究では, 遺伝子のコーディング方式として多重遺伝子を採用した. 多重遺伝子は各仕事識別子を機械数分ずつ持ち, n/m 問題における遺伝子長は $n*m$ となる. 3/2 問題における例とデコード方法を図 1 に示す.

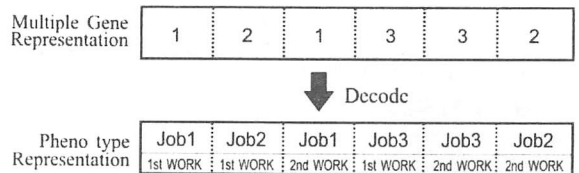


図 1 多重遺伝子のデコード例

ここで, n/m 問題において, ϕ をある多重遺伝子の評価値とする. $S_i (i=1, 2, \dots, n*m)$ を多重遺伝子の先頭より i 番目の遺伝子とし, $\phi(i, j)$ は S_i と S_j を交換した場合の多重遺伝子の評価値とする. $\phi(i \neq j)$ は S_i と S_j 間の遺伝子の並び順を逆順にした場合の多重遺伝子の評価値とする. $\Pi(i, j)$ は S_i から S_j 間の部分遺伝子群を示す. また, 近傍係数を学習回数 t の関数として $d(t) (d(t) < n/2)$ とする. ただし, $d(t)$ は定数とすることもできる.

- 以上の条件下での, LCO のアルゴリズムを以下に示す.
- (1) ランダムに実行可能スケジュールを発生し, 多重遺伝子にエンコードする.
 - (2) ランダムに S_c を決定し, $\Pi(c-d(t), c+d(t))$ を選択する.
 - (3) $\Pi(c-d(t), c+d(t))$ 内でクラスタリングにより最適化する.
 - (4) 必要ならば, $d(t)$ を変化させ, (2)へ戻る. ただし, (3)の操作で解が改善しない場合が連続で発生し, それが指定回数に達した場合は計算を終了する.
- ここで, (3)の具体的な探索アルゴリズムは示されていない. 下記に 5 つのクラスタリングアルゴリズムを提案する.

3.3. SEM (Simple Exchange Method)

- (1) $i=0$ とする
- (2) i を $i+1$ とし, $\phi > \phi(c, c+i)$ ならば, S_c と S_{c+i} を交換.
- (3) $\phi < \phi(c, c-i)$ ならば, S_c と S_{c-i} を交換.
- (4) $i < d(t)$ ならば (2)へ戻る. $i = d(t)$ ならば終了.

3.4. IEM (Inverse Exchange Method)

- (1) $i=0$ とする

- (2) i を $i+1$ とし, $\phi > \phi(c \leftrightarrow c+i)$ ならば, S_c と S_{c+i} 間の遺伝子の並び順を逆転させる。
- (3) $\phi < \phi(c \leftrightarrow c-i)$ ならば, S_c と S_{c-i} 間の遺伝子の並び順を逆転させる。
- (4) $i < d(t)$ ならば(2)へ戻る。 $i = d(t)$ ならば終了。

3.5. SM (Smoothing Method)

- (1) $i=0$ とする。
- (2) j を $j+1$ とする。 $\phi > \phi(c-d(t)+i, c-d(t)+i+j)$ ならば, $S_{c-d(t)+i}$ と $S_{c-d(t)+i+j}$ を交換。
- (3) $i+j=2d(t)$ ならば i を $i+1$ とし, $j=0$ とする。
- (4) $i < 2d(t)$ ならば(2)へ戻る。 $i = 2d(t)$ ならば終了。

3.6. SYM (Symmetrical Exchange Method)

- (1) $i=0$ とする
- (2) i を $i+1$ とし, $\phi > \phi(c-i, c+i)$ ならば, S_{c-i} と S_{c+i} を交換。
- (3) $i < d(t)$ ならば(2)へ戻る。 $i = d(t)$ ならば終了。

3.7. LCO/SI (LCO with Single Immune)

3.2 から 3.6 に示した, JSP に対する LCO の基本アルゴリズム及び探索交換アルゴリズムを用いて, JSP 近似解を導出する予備実験を行ったところ, 一度局所解に陥った場合, それを抜け出せず早期に計算が終了してしまう場合が生じることが分かった。これを解決するために, 単一免疫(記憶)細胞を持ち, 局所解から抜け出しやすくするアルゴリズムを取り入れた LCO を開発した。3.2 から 3.6 に示されるアルゴリズムは以下のとおり, 変更される。

- (1) LCO/SI を導入するために, 新たな変数を用意する。 ϕ_T をある多重遺伝子の評価値に温度係数 $T(T > 1.0)$ を乗じたものとする。 ϕ を免疫遺伝子の評価値とする。3.3 から 3.6 にある, ϕ を ϕ_T と読み替えて 3.2 に示すアルゴリズムの(4)以降を以下として LCO を実行する。ただし, $\phi(i, j)$ の定義は変更しない。
- (4) $\phi > \phi_T$ ならば, ϕ_T を ϕ とする。(免疫の更新)
- (5) 必要ならば, $d(t)$ を変化させ, (2)へ戻る。ただし, (4)の免疫の更新が一定回数連続で発生しない場合は計算を終了。

3.8. ジグ・工具割当の考慮

ジョブ i の機械 j による加工工程 O_{ij} は使用可能なジグ・工具の集合 b_{ij} を持つ。従って, 現実的なスケジューリングシステムでは, 有限個であるジグ・工具の割当を考慮する必要がある。

JSP の多重遺伝子表現に対する, LCO の適用法は上述した。3.3 から 3.6 のアルゴリズムを適用するためには, 多重遺伝子をデコードし, 実行可能スケジューリング解を求め, そのコスト ϕ を計算する必要がある。以下に, デコード方法を示す。

今, 多重遺伝子染色体 $G = \{G_i | i=1, \dots, l, \dots, n \cdot m\}$ を先頭より順次デコードする際に, 同じジョブの l 番目の工程は, ジョブ i , 使用機械 j , 使用可能ジグ・工具集合 $b_{ij} = (b_{ijk} | k=1, \dots, bn_{ij})$ を持つ。 $l-1$ 番目までの処理で, ジョブ, 機械及びジグ・工具はそれぞれ, 拘束終了時間 p_{ji}, pm_j, pb_k が決定される。 pb_k は, 式(1)により導出される。

$$b_k = \begin{cases} \arg \min_k (b_{ijk}) & : \forall kb_{ijk} > p_{ij} \\ \arg \max_k (b_{ijk}) \wedge b_{ijk} \leq p_{ij} & : \text{otherwise} \end{cases} \quad (1)$$

where $p_{ij} = \max(p_{ji}, pm_j)$

また, l 番目の工程の始期 C_l は,

$$C_l = \max(p_{ji}, pm_j, pb_k) \quad (2)$$

により決定する。式(1)及び(2)を満たすように, デコードされた多重遺伝子は実行可能スケジューリング解であり, 評価値 ϕ が計算可能である。よって, LCO 及び各クラスタリング規則が適用可能となる。

4. 数値計算実験

LCO の JSP に対する有効性を検証するため, 数値計算実験を行った。C/C++言語によるコードを VisualStudio.net の最適化でコンパイルした。実行環境は OS: Windows XP(NTS.1), CPU: Intel P4 2.26GHz, 主記憶: PC2100 DDR 1GB の PC/AT 互換機である。

実験対象は 10/5・50/6・150/6 の 3 問題であり, 各問題を乱数により生成した。評価は最大滞留時間と平均納期遅れ時間に同一の重みを乗じて計算した。また, 3.3 から 3.6 に示す 4 方法に SI を適用し, IGA(Inversion GA)との比較実験を行った。

LCO の近傍値は, $0 < d(t) < 0.4n$ を満たす乱数を 1 回ごとに発生させて決定した。終了条件は, 10/5・50/6・150/6 問題でそれぞれ 80・60・40 回とした。IGA の個体数は 30 とし, 突然変異・交叉・逆位は個体ごとにそれぞれ 10%の確率で発生させた。図 2 に LCO 及び GA の近似解の評価値比較グラフを示す。また, 図 3 に計算時間比較グラフを示す。解精度指標は, LCO で得られた近似解の評価値を, GA で得られた近似解の評価値

で除して計算した。計算時間指標も同様に比較した。図より, LCO は GA よりも高精度の近似解を計算することが分かる。計算時間は解精度が良好である SEM 及び SYM において, GA とほぼ同等であるが, これは, 早い時期に局所解収束が発生しているためであり, GA が計算速度面でのパフォーマンスが優れているとは言えない。GA で LCO 程の精度の近似解を計算するには, LCO と比較し莫大な計算時間を要する。

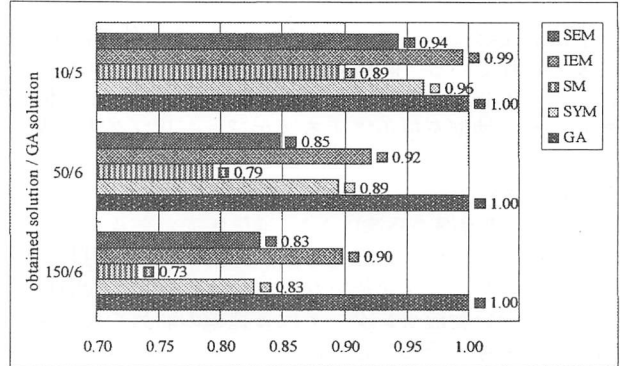


図 2 LCO 及び GA のスケジューリング解評価値比較グラフ

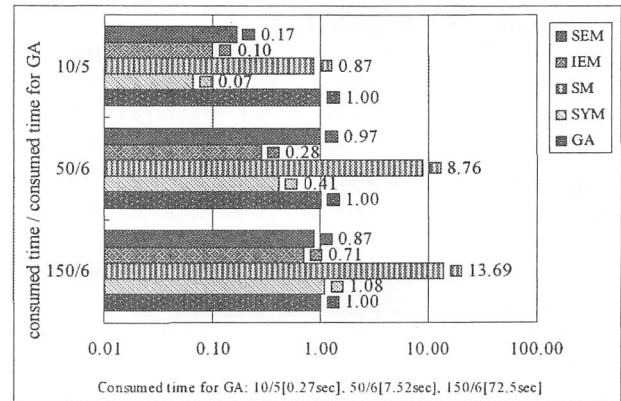


図 3 LCO 及び GA の計算時間比較グラフ

5. 考察

LCO は各問題において GA よりも高い評価値の近似解を計算する。中でも SM は最も良い解を持つが, 計算量がジョブ数と機械数の積の自乗に比例して増加するため, 中規模以上の問題には向かない。計算時間と解精度のバランスからすると, JSP に対しては SYM ないし SEM が最も有効な方法である。ジグ・工具を考慮しない JSP では, SYM・SEM は GA による近似解よりも, 中規模の問題で 10%程度, 大規模の問題では 12~14%程度良い評価値の近似解を計算したが, ジグ・工具を考慮した JSP においては, LCO の有効性が高く, GA と比較した改善率は, 同様に 11~15%及び 17~18%である。

6. 結言

TSP に対して有効な解法であることを示した LCO は, JSP に対しても有効な解法であることを検証した。このことは, LCO は 1 つの解を 1 つの一意の遺伝子によって表現できるタイプの組合せ最適化問題に対して適用可能であり, これまでの他の解法よりもより良い近似解を計算できる可能性を示唆する。また, ジグ・工具割当を考慮した JSP において, LCO は GA より良好なパフォーマンスを示した。これは, LCO のダイナミックなクラスタリング規則が, 問題の記述の複雑さに関係なく, 有効に作用するためである。

著者らが提唱する, 局所クラスタリングの繰返しによる大局的組織化の理念は TSP・JSP に対する実験により, ほぼ確立できたと考えられる。今後, この理念のもとに LCO を更に発展させ, より多くの問題に対して適用し, その有用性を検証したい。

<参考文献>

- 1) 古川 正志他;システム工学,コロナ社(2000)
- 2) RICHARD W. CONWAY 他著・関根智明他訳; スケジューリングの理論; 日刊工業新聞社 (1971)
- 3) Teuvo Kohonen; Self-Organizing Maps, Springer Verlag (1995)
- 4) 渡辺他; ジョブのジグ・機械割り当てを考慮したスケジューリングの開発, 精密工学会誌 66 巻第 8 号 PP1185-1189(2000)