二次割当問題に対する可変深度近傍探索の適用

清水 達朗（沼田 一道 教授，松浦 隆文 助教）

１． はじめに

二次割当問題（Quadratic Assignment Problem：以下 QAP）は，n 個の施設を n 個の場所へ，「施設間流量と場所間距離の積」で定義されるコストの総和を最小化するように割当てる問題である．相互関係を有する施設の配置問題として提起されて以来，代表的な組合せ最適化問題の一つとして現在でも盛んに研究されている[1]．しかし，QAP は *NP*−困難な問題クラスに属し，能率の良い厳密解法は知られていない． 問題例（群）の構造ごとに工夫・考案された分枝限定法ベースの解法が，$n = 30$くらいまでの問題例を厳密に解ける程度である[2]．ただし，（その程度の問題例でも）厳密解の計算には莫大な時間がかかるので，現実的な応用の観点からは，短時間で精度の高い解を探し出す発見的解法が求められている．

本研究では，可変深度近傍探索（Variable Depth Neighborhood Search：以下 VDS）[3]を用いた QAP に対する新たな発見的解法を提案する．VDS は，現在解に対し小さな変更を連鎖的に行なって大きな変更を実現し，連鎖の長さを動的に制御することで能率よく改善解を探索する仕組みである．以下では，QAP の問題構造をふまえた上で VDS の動作を具体化し発見的解法として構成する．これを用いて QAP のベンチマーク問題[4]を解き，既存の発見的解法による結果[3]と比較して提案解法の性能を評価する．

２． 二次割当問題(QAP)

２．１ 定義

施設の集合を $N = \{1,2,\dots n\}$，場所の集合を $N' = \{1,2,\dots n\}$ とする．施設の配置を $N(= \mathrm{N}')$ 上の置換 σ で表す．すなわち，$\sigma(i)$ で場所 i に置かれる施設を表す．N 上の置換全体の集合を S_N と書く．

$D = (d_{ij})$ および $F = (f_{k\ell})$ を，場所 i と場所 j 間の距離，施設 k と施設 ℓ 間の流量を表す $n \times n$ の行列とする．QAP は(1)(2)式を満たす σ (解)を求める問題である．

$$min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} f_{\sigma(i)\sigma(j)} \qquad (1)$$

$$s.t. \quad \sigma \in S_N \qquad (2)$$

２．２． 定式化①

場所 i に施設 k を割当てるとき１，そうでない場合に 0 をとる決定変数 x_{ik} を用いて表現する．

$$min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} d_{ij} f_{k\ell} x_{ik} x_{j\ell} \qquad (3)$$

$$s.t. \quad \sum_{k=1}^{n} x_{ik} = 1 \quad i = 1,2,\dots n \qquad (4)$$

$$\sum_{i=1}^{n} x_{ik} = 1 \qquad k = 1,2,\dots n \qquad (5)$$

$$x_{ik} \in \{0,1\} \qquad i,k = 1,2,\dots n \qquad (6)$$

(3)式は割当コストの総和（目的関数）の最小化を表している．(4)～(6)式は制約条件である．(4)式は各場所には 1 つの施設が配置されること,(5)式は各施設をいずれか 1 つの場所に配置することを表している．(6)式で，x_{ik} を 0-1 変数に規定している．

２．３． 定式化②

定式化①は，目的関数が二次式なので，このままでは汎用ソルバーで解くことができない．そこで，QAP を線形整数計画問題の形で表現するために 0-1 決定変数 $y_{ijk\ell}$ を導入する．$y_{ijk\ell}$ は「場所 i に施設

k を割当て，かつ，場所 j に施設 ℓ を割当てる」場合に 1 をとり，そうでない場合に 0 をとる 0-1 決定変数である．

$$s.t. \quad y_{ijk\ell} \geq x_{ik} + x_{j\ell} - 1 \qquad (7)$$

$$y_{ijk\ell} \leq x_{ik} \qquad (8)$$

$$y_{ijk\ell} \leq x_{j\ell} \qquad (9)$$

(7)～(9)は $i, j, k, \ell = 1,2, \ldots n$

(7)～(9)式全てで $y_{ijk\ell} = x_{ik}x_{j\ell}$ と等価である．変数の個数は定式化①の n^2 から，n^4 に増加するが，目的関数を線形で表現することができる．目的関数(10)式に制約条件(4)～(9)式を加えたものが定式化②である．

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} d_{ij}f_{k\ell}y_{ijk\ell} \qquad (10)$$

２．４．　厳密解法での数値実験

QAP のベンチマーク問題集 QAPLIB[4]にある問題例 chr15a($n = 15$)を使用した．前節で示した定式化②を用い汎用ソルバーGurobi で計算したところ，16048 秒で最適解を求めることができた（実験環境：Windows7，intel core i5 2.67GHz，メモリ 32GB)．キーボードの平均打鍵移動時間を意識した bur26a($n = 26$)などは，1 週間（！？）計算しても最適解が求まらなかった．

３．　既存の解法と本研究の解法の方針

QAP の基本となる探索法は 2-swap（1 対の施設の交換）である．発見的解法の基本戦略である局所探索法は，現在解 σ の近傍 $N(\sigma)$の中から最良解を選び次の解 σ'とする手続きの反復を軸とし，しばしば 2-swap を探索の基本とする．2-swap をメタ戦略の一種であるタブー探索[3]で強化した発見的解法は多くの QAP のベンチマーク問題に対して優れた結果を与えている[4]．

今回は，基本探索法に VDS を採用する．本研究の特徴的な点は，2-swap を拡張した形で VDS を構成することである．これは，QAP に対して高い探索能力を期待したものである．

４．　提案解法

４．１．VDS の特徴

VDS は 2-swap を連鎖的に行なって（＝複数個の施設を 1 回の交換に関係させて）大きな変更（巡回交換）を実現する．また，連鎖の長さを動的に制御すること（探索木の導入）で改善の可能性が高い解を効率良く探す．

まず初めに大きな変更である巡回交換 CycleSwap$^{(k)}$（以下 CS$^{(k)}$）を説明する．CS$^{(k)}$ は，k を 1 回の交換に関わる施設個数とすると，k 個の施設を「巡回させるように交換」することである．図 1 は CS$^{(3)}$の例で，場所 2 に施設 3 を，場所 3 に施設 5 を，場所 5 に施設 2 を巡回的に交換することを表している．

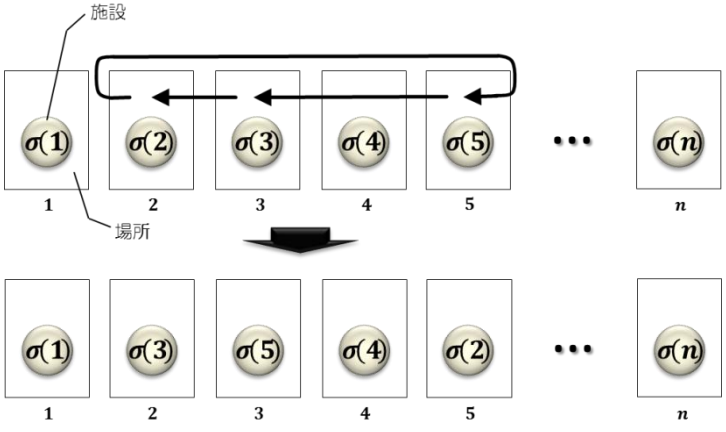

図 1：CS$^{(3)}$（$k = 3$ の巡回交換）の例

巡回交換は 2-swap を連鎖的に行うことで実現できる．図 1 の例でいえば，施設 2 に注目して 2-swap の操作（移動）を 2 回行うことで同じ結果が得られる．さ

らにいえば, これは最初に施設２を外した状態にし, それを穴埋めしていくように 2-swap を繰り返し, 最後の空きを施設２で埋めるという手続きだとわかる．すなわち巡回交換は「１つ施設が抜けた状態で, それを穴埋めしていくような施設交換を繰り返し, 最後の空きを最初に注目した施設で埋める[5]」ことで実現される．

次に，連鎖の長さを動的に制御することについて説明する．これは可変的に施設の交換個数を調節することにあたる．この制御を探索木（図２）を用いて行う．図の探索木の節点の上部は深度のレベル（$level$），下部は新たに取り外す施設を示す．$level$ は 2-swap の操作を施した回数に対応する．探索木の頂点から $Level = \alpha$ （α は 0 以上の整数）まで辿っていくと，そこまでに選択した施設による巡回交換が完了することに注意したい．

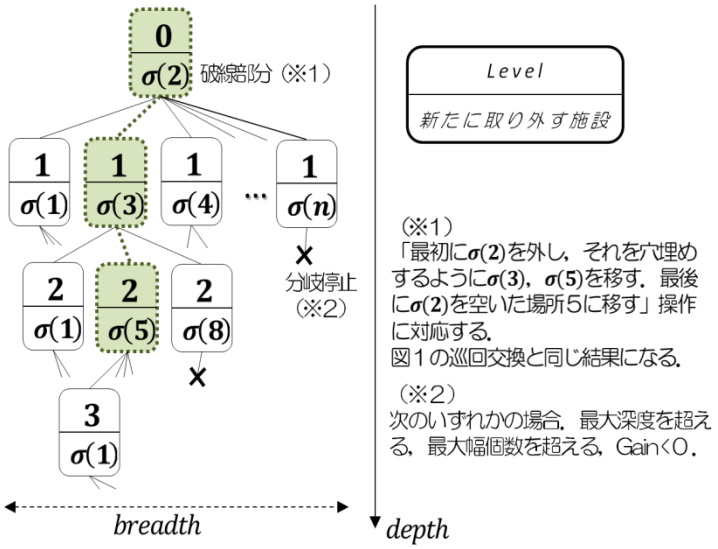

図 ２：探索木の例

探索の手順は深度優先によって行う．

まず $Level = 0$ で最初に取り外す施設を指定する（2-swap の準備段階）．最初に取り外す施設はN の全ての可能性が順に試される．それから，空白になった場所に施設を移動させる操作を行う．このとき新たに取り外す施設は，今まで選ばれていない施設の中から，なるべく目的関数値を改善するような施設を選択したほうが良い．どの施設を空白に移動させるかは，$Level = \alpha$ と $Level = \alpha + 1$ との比較を通じて行う（2-swap の妥当性判断）．

$Level = \alpha$ 時点の解をσ_α，交換後（$Level = \alpha + 1$ 時点）の解を$\sigma_{\alpha+1}$とする．探索木の $Level = \alpha$ で取り外す施設を $L^{(\alpha)}$ とする．最初に取り外した施設は $\mathrm{L}^{(0)}$ と表される．「深度を増やす」操作は，「σ_α において $\mathrm{L}^{(0)}$ と新たに交換する施設（$\mathrm{L}^{(\alpha+1)}$）」を決める操作である．$\mathrm{L}^{(0)}$ と $\mathrm{L}^{(\alpha+1)}$ の交換の妥当性を評価する式は，(11)式をもとに(12)式で定義される．

$$Cost^{(\sigma,\sigma(i))} = \sum_{j=1}^{n} \mathrm{d}_{ij} f_{\sigma(i)\sigma(j)} + \sum_{\substack{j=1 \\ i \neq j}}^{n} \mathrm{d}_{ji} f_{\sigma(j)\sigma(i)} \qquad (11)$$

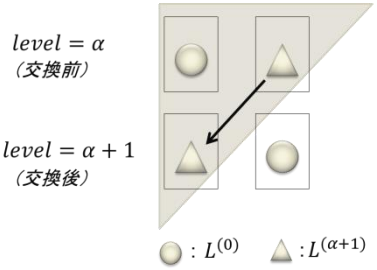

図 ３：「深度を増やす」妥当性の評価

(11)式は「現在解が σ のとき, 施設 $\sigma(i)$ の割当てに必要なコスト」を表す（$i \neq j$ はコスト式の二重カウントを防いでいる）．

$$Gain^{(L(\alpha+1))} = \mathrm{Cost}^{(\sigma_\alpha, \mathrm{L}^{(1)})} + Cost^{(\sigma_\alpha, \mathrm{L}^{(\alpha+1)})} - \mathrm{Cost}^{(\sigma_{\alpha+1}, \mathrm{L}^{(\alpha+1)})} \qquad (12)$$

(12)式で交換前と交換後のコストの差分（$Gain$）を評価する（図３の灰色で塗りつぶしたところ）．$Gain$ が大きい施設（図３の△）ほど，移動させる価値がある．$Gain < 0$ ならば，交換後にコスト総和は増加するので, その交換は採用しない. 施設移動が不可能な場合は１つ深度を戻す. 移動ごとに, 巡回交換を完成させコストを調べる必要がある．このような条件の下，交換の一進一退を繰り返しな

がら良い巡回交換を探す.計算量の観点から,最大深度(定数)と各深度毎に最大幅(定数)を設定する.

４．２．　解法

Step1：初期解σをランダムに作成.

Step2：巡回交換に関わる施設のリスト$[L^{(0)},L^{(1)},..,L^{(n)}]$の初期化．$L^{(0)}=1;\ \alpha=1;$

Step3：$\alpha>0$の間，Step3-a もしくは Step3-b を繰り返す.

Step3-a：深度α増加の手続き．深度αにおける$Gain$のリスト作成．$Gain$のリストを降順にソートし，$Gain$が大きく未探索の施設の交換を実施．コスト計算．改善時は，即座に解を更新し，Step2 へ戻る.

Step3-b：深度αが(において)最大深度(幅)を超えたとき，$Gain<0$ のときは，深度α減少の手続き.

Step4：$L^{(0)}$++；$\alpha=1;\ L^{(0)}\le n$ ならば，step3 へ戻る．$n<L^{(0)}$ ならば，VDS 終了につき，解の出力.

Step5：Step1～4 を 10～100 回程度繰返し(多スタート探索法)，最良解を出力.

５．　数値実験

Borland 社の Delphi6 で提案解法を実装した.問題データは QAPLIB から選択し数値実験した(表１).最大深度5，最大幅は深度順に n,n,n,5,5,5 とした．chr15a，nug30 は最適解が知られており，tai80a，sko90 はタブー探索に基づく解法が現在最も良い解を導いている.

結果は，交換近傍よりも良い解を発見できた.chr15a や nug30 は 10 通りの初期解では最適解を導けなかったが 100 通りの初期解では最適解を導くことができた．表以外の問題例でも，$n\le 30$ では最適解を導くことがあった.

表 1：結果（誤差(%)は最も良かった解のもの）

問題名	10通りの初期解 交換近傍, BestMove 誤差(%)	10通りの初期解 VDS 誤差(%)	100通りの初期解 VDS 誤差(%)	実験1回あたり VDS 平均実行時間(ms)
chr15a	23.88400	0.40420	0.00000	58
nug30	2.25343	0.29393	0.00000	6457
tai80a	3.03751	1.52485	1.51288	948486
sko90	1.48181	0.66993	—	50582076

注：sko90は計算に時間がかかった為，100通りの初期解の測定はしていない.

しかし，n が大きくなると実行時間がかなり増大することがわかる．理由に，(1)式で表されるコスト計算に時間がかかることが挙げられる．探索木の最大幅を設定して余計なチェックを減らし，最大深度を制限することで見込みの低い探索をしないようにはしているものの，表のように実行時間が指数関数的に増加してしまった.高速化を考えるなら，n を減らす，5 を少なくするなどが考えられる.

６．　まとめ

VDS の動作を具体化し QAP のベンチマーク問題に適用した．結果より，可変深度近傍による探索が QAP に対して有効な枠組みであることがわかった.単純な交換近傍よりも良い探索が可能で,$n\le 30$ では最適解を短時間で見つけることもあった．$n>30$ ではある程度の解を導くことができるが，実行時間の観点から，探索木の最大深度と幅を調節する必要がある．今後の課題は，深度の増減時の評価式をさらに QAP の問題構造にあわせたものにすることである.

参考文献

[1]久保幹雄，田村明久，松井知己編(2002)，応用数理計画ハンドブック，朝倉書店，1354pp.

[2]藤澤克樹，梅谷俊治(2009)，応用に役立つ 50 の最適化問題，朝倉書店，134pp.

[3]柳浦睦憲，茨木俊秀(2001)，組合せ最適化－メタ戦略を中心として－，朝倉書店，237pp.

[4]QAPLIB，http://www.opt.math.tu-graz.ac.at/qaplib/inst.html (2013.12.25).

[5]沼田一道，濱川大志(2008)，自動倉庫の入出個スケジューリング問題に対する可変深度近傍を用いた発見的解法，システム制御情報学会論文誌，21(5),139-141.