1996年度日本オペレーションズ・リサーチ学会
春季研究発表会

再配置問題に対する線形時間移動手順決定法

01605520 NTT 通信網研究所 ＊巳波弘佳 MIWA Hiroyoshi
01009550 NTT 通信網研究所 伊藤大雄 ITO Hiro

1．まえがき

再配置問題とは，容量を持った複数の倉庫と，それらに収容されている他の倉庫に移動すべき複数の荷物があるとき，与えられた移動回数以下で，全ての荷物を目的の倉庫に移動させる問題である．ただし，倉庫に収容されている荷物の大きさの和は，常にその倉庫容量を越えてはならない．

再配置問題に対しては既に，すべての荷物の大きさが1の場合，すべての荷物が移動可能か否かを判定することは線形時間で可能であることが分かっている[1]．しかし，荷物の移動手順を求める線形時間アルゴリズムは得られていなかった．実際，荷物数を m とすると，[1] の証明で用いられた方法に従えば，$O(m^2)$の計算時間が必要であった．本稿では，荷物の移動手順も線形時間で求めることができるアルゴリズムを提案する．

2．再配置問題

2．1 諸定義

倉庫と荷物をそれぞれ節点と枝に対応させることにより，再配置問題を，グラフを変形する問題として定式化できる．

【定義1】

有向多重グラフ$G=(V, E)$（ただし節点集合 V ($|V|=n$)，枝集合E ($|E|=m$)，それぞれ$V(G)$，$E(G)$とも表す）．枝eの始点をs_e，終点をt_eと表し，枝eを(s_e, t_e)とも表す． □

【定義2】

強連結成分間に，次のように半順序関係‘>’を定義する．つまり，異なる二つの強連結成分C_iとC_jの間に，始点を$u \in V(C_i)$かつ終点を $v \in V(C_j)$とする枝(u, v)が存在するとき，$C_i > C_j$とする． □

次に，再配置問題を定義する．まず，ネットワーク$N=(G, c, d)$は，有向多重グラフ$G=(V, E)$，節点容量集合$c=\{c_v | v \in V\}$，枝容量集合$d=\{d_e | e \in E\}$からなる．節点$v \in V$の空き容量b_vとは$b_v = c_v - \sum_{\forall e \in E, s_e = v} d_e$で定義され，節点部分集合$S$の空き容量とは．$S$に含まれるすべての節点の空き容量の和で定義される．ネットワークNに対する操作$Move(e)$ ($e \in E$)は以下のように定義される．まず，Eから枝eを取り除き，Eに$s_{e'} = t_{e'} = t_e$を満たすループe'を付け加え，$d_{e'}$と$W_{e'}$をそれぞれ$d_{e'} = d_e$，$W_{e'} = W_e \cup \{s_e\}$とする．節点$t_e$の空き容量が$d_e$以下ならば操作$Move(e)$は実行可能ということにする．ネットワークのグラフの枝を全てループにするような$Move$からなる操作列で，含まれる操作$Move$の数が与えられた正の自然数H以下ならば，Nは実行可能なネットワークであると呼び，操作列を実行可能操作列と呼ぶ．

【定義3】（再配置問題）

INSTANCE: ネットワーク$N=(G, c, d)$，正の自然数H．ただしすべての節点$v \in V(G)$に対して $c_v \geq \sum_{\forall e \in E, s_e = v} d_e$を満たしており，$E(G)$はループを含まない．

QUESTION: Nは実行可能なネットワークか？ □

【定義4】（**Vacancy Rule**）

$$c_v \geq \sum_{\forall e \in E, s_e = v} d_e, \quad c_v \geq \sum_{\forall e \in E, t_e = v} d_e \quad (\forall v \in V(G))$$

□

2．2 従来の結果

荷物の大きさがすべて1の場合に対応する再配置問題については，次の定理が証明されている[1]．

【定理1】

荷物の大きさがすべて1の場合に制限した再配置問題について

(i) ネットワーク$N=(G, c, d, W)$ ($^\forall e \in E, d_e=1, W_e=\phi$)が実行可能であるための必要十分条件は，$N$が次の(a)かつ(b)（または(b')）を満たすことである．

(a)NはVacancy Ruleを満たす．

(b) Gの節点数2以上の任意の連結成分の空き容量は1以上ある．

(b')Gの定義2の意味での強連結成分の極小元に少なくとも1以上の空き容量を持つ節点が存在する

(ii)ネットワーク$N=(G, c, d, W)$ ($^\forall e \in E, d_e=1, W_e=\phi$)が実行可能かは線形時間で判定できる． □

以上の結果からは，実際に移動させる順序を線形時間で決定することはできなかった．実際，ネットワークのグラフの極小元の空き容量が1以上の節点を終点とする枝に対して操作$Move$を適用していけば，実行可能な操作列を得ることができるが[1]，一度操作$Move$を適用するごとに，ネットワークの極小元を強連結成分分解を行って求めなければならないので，この方法を忠実に実行すると計算量は$O(m^2)$である．

3．線形時間アルゴリズム

本節では，荷物の大きさがすべて1の場合，すべての荷物を移動させる順序を線形時間で求めるアルゴリズムを提案する．実行可能性の判定は定理1より線形時間で可能であるので，本節では入力は実行可能なものに限ることとする．また，操作$Move$を一度適用してループ化された枝に対しては二度と操作が加えられることはないので，本節では，操作が加えられた枝は除去されるものと

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

する．

提案アルゴリズムは，深さ優先探索を行い，バックトラックを行う際に当該枝に操作*Move*を適用していく方法を基本とする．以下でもう少し詳しく説明する．

実行可能ネットワーク$N=(G, c, d, W)$ ($^{\forall}e\in E, d_e=1, W_e=\phi$)が与えられたとする．すべての枝に'*unscanned*'のラベルを付与し，'*unscanned*'の枝の始点であるすべての節点のラベルを'*unfinished*'とし，'*unscanned*'の枝を始点に持たない節点のラベルを'*finished*'とする．任意の'*unfinished*'のラベルの付いた節点vから手続きsearch(v)で枝の探索と操作*Move*の処理を行う．

次に手続きsearch(v)を説明する．まず，探索された枝を格納するための双方向リストPを用意し，初期状態は空集合とする．次に，節点vにポインタptを置き．以下の処理を行いながらptを移動する．ptが置かれた節点wにおいて，

(I)「節点wのラベルが'*unfinished*'である」場合：　節点wを始点とする'*unscanned*'のラベルがついた枝のうち一つを，Pの最後に加えて枝のラベルを'*scanned*'に付け替える．この時，もし節点wを始点とする'*unscanned*'のラベルがついた枝がなくなれば，節点wのラベルを'*finished*'に変更する．　最後に，ptを枝の終点に移動する．

(II)「節点wのラベルが'*finished*'であり，節点wの空き容量が1以上あり，Pが空集合でない」場合：　Pの最後の枝eに実行可能な操作*Move*(e)を適用した後，Pから枝eを取り除き，ptをs_eに戻す．

(III)「節点wのラベルが'*finished*'であり，節点wの空き容量が0であり，Pに含まれる枝に，その始点のラベルが'*unfinished*'であるか，空き容量が1以上であるものが存在する」場合：　この時，Pに含まれる枝はサイクルを構成している．なぜなら，(a)Pの枝は有向パスを構成していることと，(b)節点wの空き容量が0なので，Vacancy Ruleより節点wを始点とする枝と終点とする枝の本数が同じでなければならず，節点wはPの最初の枝の始点でもなければならない，ということから分かる．そこで，リストPを$P=<e_1, e_2, \ldots, e_{p-1}, e_p, \ldots, e_{r-1}, e_r>$（ただし<, >は順序付けられた集合を表すとする）とした時，Pの最後の枝e_rから逆方向に探索し，条件を満たす最初の枝e_pの始点v_pにptを移動し，$P=<e_p, \ldots, e_{r-1}, e_r, e_1, e_2, \ldots, e_{p-1}>$と変更することによって$P$を再構成する（図1参照）．$pt$が移動する節点$v_p$のラベルは'*unfinished*'であるか，空き容量が1以上であるので，次のステップでは必ず (I) または (II) の条件が満たされる．

(IV)「節点wのラベルが'*finished*'であり，Pが空集合である」ならば，手続きsearchを終了する．

ptが置かれた節点wにおいては，「節点wのラベルが'*finished*'であり，節点wの空き容量が0であり，Pに含まれるすべての枝の始点のラベルが'*finished*'であり，かつ空き容量が0である」という場合はあり得ない．なぜなら，この場合もPに含まれる枝はサイクルを構成するが，このサイクル上の節点を始点とするラベル'*unscanned*'の枝がないことから，強連結成分の極小元となっている．サイクルのどの節点の空き容量も0であることから，定理1の (i) よりネットワークNは実行不可能であり，前提に矛盾する．従って，ptが置かれた節点においては，上のI,II,III,IVの場合しかありえない．

手続きsearch(v)が終了し，なお'*unfinished*'のラベルの付いた節点が存在する場合は，そのうち一つの節点を選んでvとおき，再び手続きsearch(v)を行う．

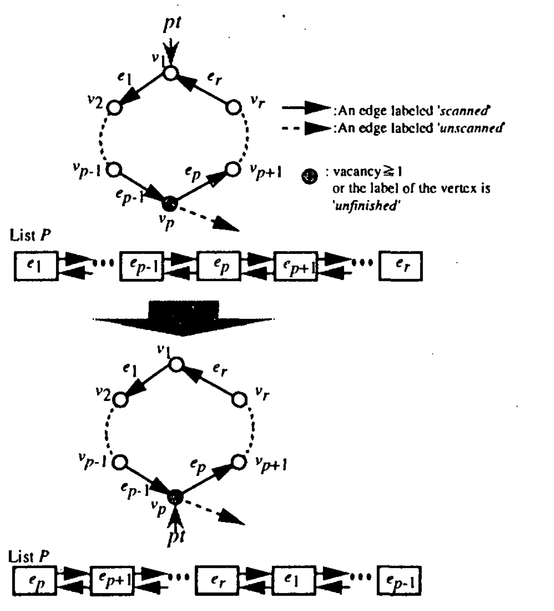

図1　ポインタptが(III)を満たす場合の処理

【定理2】

実行可能ネットワーク$N=(G, c, d, W)$ ($^{\forall}e\in E, d_e=1, W_e=\phi$)が与えられた時，上記のアルゴリズムは，$N$の実行可能操作列$S$を$O(m)$時間で計算する．

（証明の概略）

アルゴリズムの正当性は，ptが指し示す節点においては常に必ずI, II, III, IVのいずれかを満足し続け，I, IIでは必ず枝のラベルを替えるか*Move*が適用されるので，すべての枝に対して必ずラベルが替えられ，その後必ず操作*Move*が適用されて停止することから分かる．$O(m)$時間で計算できることは，各I, II, III, IVが，各枝に対して定数回しか適用されないことから分かる．　□

4．まとめ

本稿では，荷物の大きさがすべて1である再配置問題に対し，荷物の移動手順を線形時間で求めることができるアルゴリズムを提案した．

参考文献

[1]Miwa, Ito, "Complexity and Algorithm for Reallocation Problem", IEICE Trans. on Funds., Vol.E79-A, No.4, 1996. (to appear)

[2]M. Garay and D. Johnson, "Computers and intractability", W. H. FREEMAN AND COMPANY, San Francisco, 1978.

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．