日本オペレーションズ・リサーチ学会
2004年春季研究発表会

レジスタ改名機構とクラス P の巡回セールスマン問題

静岡大学情報学部　　*秋川　友宏　　HARAIKAWA Tomohiro

レジスタ (一時記憶域) の割付問題は，計算機の実行性能を左右する NP 困難な問題である．スライド/ローテイトレジスタなどの改名機構により割付問題には TSP の構造が現れるが，この TSP は意外にも NP ではなく P で解くことができる．

1　あるホテルマンの困難

本稿で扱う“レジスタ割付問題”の数学的な性質は，ある種のホテルになぞらえて説明するとわかりがよいと言われている．どんなホテルかここでつまびらかに述べることは大変はばかられるので割愛するが，とにかく，このホテルの利用客はご休憩利用とご宿泊利用とに分類することができる．

1 日を 3 時間ごとに区切って毎日の利用状況をグラフにしたものが図 1 右のようであり，v_1 から v_8 までの 8 組のカップルが毎日決まって同じ時刻にこのホテルを足しげく利用しているものとする．時間帯の重ならない利用客には同じ部屋を使い回せるので，この場合，ホテルは 8 室以下で常連客をさばけそうである．ではいったい何部屋あれば必要十分か，というのが問題で，これがレジスタ割付問題と数学的に関係する．

各時刻 t (図 1 右では $0 \le t < 8$) においてホテルに滞在しているカップルの組数をグラフ G の時刻 t における幅とよび，$width(G,t)$ と書くことにする (具体的な幅は図 1 に示した)．1 日を通じた $width(G,t)$ の最大値は $W_{max}(G)$ と書く (同様に，最小値は $W_{min}(G)$)．部屋数は $W_{max}(G)$ ぶんあれば足るだろうという予想があるかもしれないが，これはあたらない．図 1 の例では $W_{max}(G)=4$ であるが，どう組み合わせてみてもこれらの常連客を 6 室以下には収容できない．

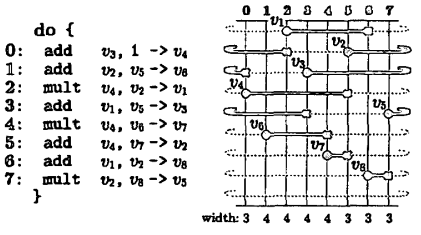

図 1　ループプログラム (左) と 循環区間グラフでの表現 (右)

ところで，必要な部屋数 R が W_{max} を下回ることがないのは自明であるから，R は非負整数 α を用いて $R = W_{max}(G)+\alpha$ と書ける．この α について，次の性質が知られている．

定理 1 ご休憩利用の常連客しかいないホテルでは，α を必ず 0 にできる．また，$\alpha = 0$ を与える具体的な部屋割りもクラス P の計算量で求まる [1]. □

定理 2 一方，日をまたぐ，すなわちご宿泊利用の常連客も混在するホテルでは，α は $\alpha \le W_{min}$ の範囲に必ず存在する．しかし，最小の α を与える具体的な部屋割りを求めることは NP 困難である [1]. □

常連客の増加に伴って部屋を効率利用する必要が生じ，ホテルマンはやがて最適解をスーパーコンピュータで求めるようになったが，常連客が 100 組に近くなろうというところでいよいよ答が出せなくなってきた．問題はご宿泊利用である．“もし夜中の 0 時にいちど作業を中断して部屋を移ってもらえるなら，定理 1 よろしく問題は簡単になるのだが...”

2　レジスタ割付問題との関係

レジスタとは，計算機の CPU の内部にある，演算用の超高速な一時記憶域のことである．限りあるレジスタを使い切ってしまうと，CPU の外部にあるメモリに途中結果をいちど退避しなければならず，プログラム，特に反復計算の実行速度が低下してしまう．高速なレジスタを効率的に使い回せる能力は，実行コードを生成するコンパイラの重要な案件である．

2.1　レジスタ割付問題

図 1 左のように，v_1 から v_8 までの 8 変数を含む反復構造のプログラム片があるとする (add は加算，mult は乗算を意味するが，気にしなくてよい)．v_1 にはステップ 2 の命令で値が代入され，ステップ 6 の命令で最後に参照されている．このとき，v_1 の生存区間は $[2,6)$ であるといい，$v_1 = [2,6)$ という半開区間で表す．v_2 は繰り返しをまたいで使われる変数であり，同様の記法で $v_2 = [5,2)$ と表す．少ない一時記憶域を使い回してこれらの値を収容せよという問題は，先ほどの部屋割り問題とまったく同じであることがわかる．

2.2　レジスタ改名機構

ところで，筑波大学が設計し，かつて世界最速を誇ったスーパーコンピュータ CP-PACS は，スライドレジスタというレジスタ改名機構を備えていた [3]．これは，図 2 左のように，実行中のあるタイミングでレジスタの内容をいっせいに別なレジスタに移せる機構である．この機構はメモリの低速性を隠蔽する目的で導入されているのであるが，本稿はレジスタ改名機構の数学的側面について報告するものであるのでアーキテクチャ論には深入りしない．なお，Intel の最新プロセッサ Itanium もローテイトレジスタという同種の機構を持つ．

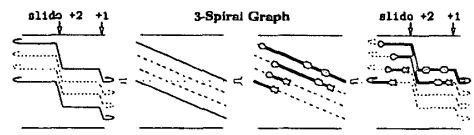

図 2　らせんグラフによるレジスタ番号のずれの表現

このような特殊機構の導入により，レジスタ割付問題は巡回セールスマン問題およびその変種と等価になる．仮に繰り返

し1回あたりのレジスタのずらし量 K を1とすると，これは1重らせんの上に生存区間を最適な順序で巻き付け，らせんの周回数を最適化せよという問題に等しいからである．たとえば，**図3右**のように $v_4 \to v_2 \to \ldots \to v_7$ の順に拾って巻き付ければ，6室必要だった部屋が4室で足ることがわかる．

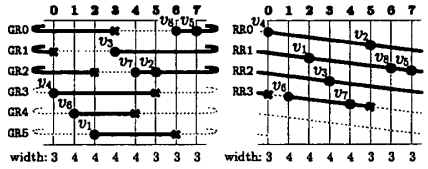

図3　従来機構 (左) と 改名機構 (右) でのレジスタ数の比較

"これで α を**必ず**小さくできるのなら部屋不足は解消できる．夜の0時にドアの部屋番号を外からいっせいに付け換えるだけでいい"，ホテルマンは喜んだ．"しかし，巡回セールスマン問題は相変わらず NP 困難．巻き付けグラフをどう最適化したものか..." $\cdots$ **ホテルマンの困難は尽きない**．

3　レジスタ改名機構とその数学的性質

ホテルマンの興味は，従前の機構と比べた α の大きさと，α を最小化するための計算量である．個々の生存区間の長さは固定であるから，α を最小にするには，生存区間どうしの**隙間** (gap とよぶ) の和を最小にするような巻き付け順を決め，らせんの周回数をできるだけ減らすしかない (**図4**)．

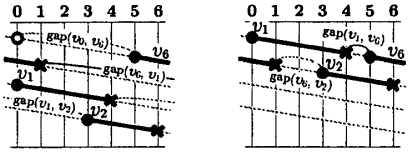

図4　生存区間どうしの距離の定義

ここで，都市間の距離に相当する生存区間どうしの隙間 gap は，グラフの横方向の広さを II として以下のように表せる．

定義 1 変数 $v_i = [s_i,\ e_i)$ と $v_j = [s_j,\ e_j)$ において，$\mathrm{gap}(v_i,\ v_j) = (s_j - e_i) \bmod II$ と定める． □

このとき，筆者らにより次が証明されている．

定理 3 $K = 1$ のとき，最後に巻き付けた生存区間に対し，それとの隙間が最も小さい生存区間を次々と巻き付ける強欲算法で，$\alpha \leq 1$ を保証できる [2]． □

このことは，必要なレジスタ数 R の上界が従来の機構よりも低く抑えられることを意味している．さらにその後，筆者らは次の定理を証明した．

定理 4 $K = 1$ のとき，α を最小化する生存区間の巻き付け順はクラス P の問題であり，その計算量は生存区間の数を N とすれば高々 $O(N^3)$ である． □

この結果は，これまでずっと NP 困難であったレジスタ割付問題が，改名機構の導入により多項式時間に落ちてきた点で非常に興味深い．**定理4**は，もちろん次のようにも書ける．

定理 4' **定義1**の非対称な距離関数を持つ N 都市の巡回セールスマン問題は，高々 $O(N^3)$ の計算量で解ける． □

意外にも，**ホテルマンの心配は杞憂に過ぎなかったのである**．

ところで，先述のとおり，改名機構はそもそもレジスタ割付問題を簡単にするために導入されたものではない．従って，K は任意に選択できず，他の制約で外的に決定されてしまう．$K \geq 1$ という一般系を考えたとき，この巡回セールスメン問題は**図5**なる制約系で表せる [3] ことまでは示せているが，この計算量が P なのか NP なのかはまだ誰にも示せていない．

$$\text{minimize} \sum_{k=1}^{K} \left(z_0 - II(1-\delta_k)\right) \tag{1a}$$

$$\text{subject to} \sum_{i=0}^{N}\sum_{j=0}^{N} (c_{ij}\,\xi_{ijk}) = z_k,\ k = 1,\ldots,K, \tag{1b}$$

$$z_k \leq z_{k-1},\ k = 2,\ldots,K, \tag{1c}$$

$$z_0 - z_k + \delta_k \geq 1\ k = 1,\ldots,K, \tag{2a}$$

$$\delta_k \in \{0,1\},\ k = 1,\ldots,K, \tag{2b}$$

$$\sum_{k=1}^{K}\sum_{i=0}^{N} \xi_{ijk} = 1,\ j = 1,\ldots,N, \tag{3}$$

$$\sum_{i=0}^{N} \xi_{ipk} - \sum_{j=0}^{N} \xi_{pjk} = 0, \quad k = 1,\ldots,K,\ p = 0,\ldots,N, \tag{4}$$

$$\sum_{j=1}^{N} \xi_{0jk} = 1,\ k = 1,\ldots,K, \tag{5}$$

$$y_i - y_j + N\sum_{k=1}^{K} \xi_{ijk} \leq N-1, \quad i \neq j = 1,\ldots,N, \tag{6}$$

$$\xi_{ijk} \in \{0,1\},\ \forall i,j,k, \tag{7}$$

$$y_i \text{ arbitrary.}$$

図5　スライドレジスタ割付問題の整数計画法での表現

現段階では，この問題を解く傍ら，条件分岐等への対応を含む近似割付の評価を行っており，追って結果を公表予定である．

謝辞　本研究は，文部科学省科学研究費補助金 若手研究 (B) 14780207 の支援を受け進められている．ここに謝意を表す．

参考文献

[1] L. J. Hendren, et. al. *A Register Allocation Framework Based on Hierarchical Cyclic Interval Graphs*, pp. 176–191. No. 641 in LNCS. Springer-Verlag, 1992.

[2] 秋川友宏 他．レジスタ割付からみたスライドウィンドウアーキテクチャの優位性について．情報処理学会第55回全国大会論文集 (1), pp. 16–17, 1997.

[3] 秋川友宏, 他．スライドレジスタ割付の厳密解法．情報処理学会論文誌, Vol. 40, No. 9, pp. 3524–3536, 1999.