解説

## 数理計画の理論と実装

# 容量制約付き車輛経路選択問題の解法

Leslie E. Trotter，大山　達雄

## 1. はじめに

1959年にDantzigら（Dantzig-Ramser[10]）によって最初に提起された車輛経路選択問題（Vehicle Routing Problem（VRP））の解法について考える．顧客の集合を $N=\{1,\cdots,n\}$ とし，それぞれの顧客 $i$ は何らかの単一品種物資に対する需要 $d_i$（整数値）を有するとする．このとき，$k$ 台の同一容量 $C$ を有する車輛を用いて，すべて中央に位置するデポ $\{0\}$ から出発するという条件下で，それぞれの顧客の物資需要を最小の総輸送費用でまかなうような配送計画を作成するにはどうすればよいか，というのが車輛経路選択問題である．顧客 $i$ から顧客 $j$ への移動コスト値を $c_{ij}\geq 0$ とするとき，すべての顧客の需要をまかなうという条件下において車輛の総移動コストが最小となるような配送計画を作成することが必要とされる．ここで $c_{ij}=c_{ji}$ が成立するとき，コストは対称であるといい，さらに同一顧客間の移動コストは $c_{ii}=0$ とする．

車輛経路選択問題を組合せ論的に述べると，集合 $N$ を $k$ 個の経路 $\{R_1,\cdots,R_k\}$ に分割するに際して，それぞれが $\sum_{j\in R_i}d_j\leq C,\ i=1,2,\cdots,k$ を満たすように，そしてそれぞれの経路に対しては配送順序を与える順列（ツアと呼ぶ）$\sigma_i$ を求める問題であるということができる．またこの問題をグラフ論的に述べると，頂点集合 $N\cup\{0\}$，枝集合 $E$，そして枝 $(i,j)\in E$ 上の移動コストを有する完全無向グラフに対して，デポ頂点を共通の頂点とする $k$ 個の閉路の和集合を解とするものを求める問題となる．ここで，それぞれの閉路は $k$ 台の車輛の各々が物資の配送を実施するルートに対応する．この問題の整数計画問題による定式化は，以下のように与えられる．

$$\text{Minimize}\quad z=\sum_{e\in E}c_e x_e$$

$$\text{subject to}\quad \sum_{e=\{0,j\}\in E}x_e=2k \tag{1}$$

$$\sum_{e=\{i,j\}\in E}x_e=2\quad \forall i\in N \tag{2}$$

$$\sum_{\substack{e=\{i,j\}\in E\\ i\in E,j\notin S}}x_e\geq 2b(S)\quad \forall S\subset N,|S|>1 \tag{3}$$

$$0\leq x_e\leq 1\quad \forall e=\{i,j\}\in E,\ i,j\neq 0 \tag{4}$$

$$0\leq x_e\leq 2\quad \forall e=\{0,j\}\in E \tag{5}$$

$$x_e：整数\quad \forall e\in E. \tag{6}$$

顧客集合 $S$ の需要を満たすのに必要とされる車輛台数の下限値 $b(S)=\lceil(\sum_{i\in S}d_i)/C\rceil$ を定義するより強力な不等式は，容量 $C$ を有するビンに集合 $S$ の需要を満たすという，いわゆるビンパッキング問題（Bin Packing Problem（BPP））に対する解を計算することによって得られる．制約条件(1)と(2)は次数制約と呼ばれる．容量制約(3)は行商人問題（Traveling Salesman Problem（TSP））に対するサブツア削除制約の一般化であると見なすことができる．これらの制約は解の連結性，そしてまたルート上の総需要が $C$ を超えないようにするためのものである．

VRPがBPPとTSPという二つの難しい組合せ問題と密接に関連していることは明らかである．$C=\infty$ のとき，多重行商人問題（Multiple Traveling Salesman Problem（MTSP））の問題例が得られるが，MTSPは頂点0と付随する枝を $k-1$ 個追加することによって等価なTSPの問題例に変換することができる（$k$ 個のデポの間には枝が存在しないことに注意されたい）．他方，与えられたVRPの問題例に対して実行可能解が存在するか否かという問題はBPPの問題例である．この問題の決定変数は，概念的には，任意の $i,j$ に対して $c_{ij}=0$ であるようなVRPモデルと等価である（すべての実行可能解は最適解である）．全体問題に対する実行可能解はTSPの（拡大グラフ

レスリー・E・トロッター
School of OR/IE, College of Engineering
Cornell University, Ithaca, NY14853 U. S. A.
おおやま　たつお
政策研究大学院大学
〒162-8677 新宿区若松町 2-2

における）ツアであって，$k$ 個のデポを連結するセグメントに沿っての総需要は $C$ を超えることはない．

BPP と TSP という二つのモデルの間には相互関連があるために，VRP の問題例は実際に解くのが非常に難しくなっている．このことは，コスト構造がルーティングの順番に大きく依存するため，それらをまとめること（パッキング）の重要性が低下していることによるものである．したがって，ルーティングとパッキングという二つの条件が競合することも時々あるため，これらのモデルの一方あるいはもう一方を緩和するという分割型最適化を用いることも必要となる．ここで，TSP 多面体上で最適化を行うという既知の手法を用いることによって BPP 構造から TSP 部分のみを抽出するという方法について調べてみよう．このことは容量制約あるいは他の妥当不等式制約のクラスに対する分離ルーティンを実施することに相当し，この分離手法は，二つのモデル間の共通部分に対応するような組合せ最適化問題に対してもより一般的に適用可能な手法である．

以下にこのアプローチの概略を紹介し，応用ソフトの SYMPHONY 構成（SYMPHONY については User's Guide[19] あるいは文献[13] を参照されたい）を用いて行った並行枝，カット，コストに対する数値実験の計算結果を示す．

## 2. 容量制約の分離

VRP 多面体に対する妥当な不等式の多くのクラスについてはいろいろな文献[1，4～7，9，14～16]に報告されているが，分離問題はほとんどの既知のクラスに対しても解くのが困難とされている．有効に分離が行われるようなクラスでも枝とカットに関しては有効とならない場合が多い．ここでは容量制約(3)を用いた VRP 多面体から任意の分数点を分離するという問題に注目する．このような制約に対する分離問題は Harche-Rinaldi（文献[5]参照）によって $NP$-完全であることが示されたが，ここに示した定式化の LP 緩和問題も構造的に $NP$-完全であることが分かる．

分離問題を解くのは非常に難しいため，制約条件を効率的に分離するための発見的解法をどのようにして構築すればよいかという問題に対する研究努力がこれまでかなり活発に行われてきた．しかしながら，Augerat *et al.*[5] による論文が現れるまで，ほとんどの既知の解法は容量制約を満たさないような条件を識別するのに有効とはいえないものであった．文献[4]では，これらの制約条件（すなわち $b(S)=(\sum_{i\in S}d_i)/C)$ の小数形式が多項式時間で分離可能であることが示されている．この方法は，式(3)の制約条件を満たさないものを識別する発見的解法として利用することができる．

前述のモデルでは TSP は拡張グラフ上の VRP の緩和形であると見なすことができるが，TSP ツアは，BPP に対する実行可能解となる場合には VRP の実行可能解にもなり得る．そこですべての TSP ツアの付随ベクトルの凸包である TSP 多面体を $T$，VRP の実行可能解に対応するツアの付随ベクトルによって生成される多面体を $R$ と表すと，$R\subseteq T$ であって，さらに $R$ の端点が $T$ の端点に含まれることが分かる．

枝カット探索木のある頂点において，LP ソルバーによって LP 緩和問題に対する最適解 $\hat{x}$ が得られたとする．このとき上の $\hat{x}$ に対応する支持グラフあるいは分数グラフを，枝集合 $\hat{E}=\{e:\hat{x}_e>0\}$ を有するグラフ $\hat{G}=(N\cup\{0\},\hat{E})$ と定義する．いま $\hat{x}$ が整数値であるとすると，$\hat{x}\notin T$ のとき，次数制約が LP 緩和問題の中に明示的に含まれているので，グラフ $\hat{G}$ は非連結となる．この場合，デポを含まないようなグラフ $\hat{G}$ の任意の連結成分の頂点集合は容量制約を満たさないので，われわれは CUT 操作を実行することによって，この不等式を LP に追加した後，再び最適化を実行する．他方，$\hat{x}\in T$ の場合，$\hat{x}\in R$ か否かを考慮し，そうでない場合には $\hat{x}$ に対応するツアのデポ間ルート中にある頂点によって導出される容量制約を $\hat{x}$ が満たさなくなるので，CUT 操作を実行する．最後に，$\hat{x}\in R$ のとき，$\hat{x}$ は VRP に対する実行可能解を与えるので，現在の探索頂点の操作は終了する．このようにして $\hat{x}$ が整数値でない場合の処理を行うことになる．

### 2.1 発見的解法

分離問題は解くのが難しいため，解 $\hat{x}$ が満たさない容量制約を決定するためにいくつかの単純な発見的解法を適用する．これらの発見的方法は支持グラフ $\hat{G}$ に対して適用されるが，ここではグラフ $\hat{G}$ は連結とし，成分が $w_e=\hat{x}_e$ によって定義されるような枝の重みを有するベクトル $w\in IR^{\hat{E}}$ を有するとする．

$$w(F)=\sum_{e\in F}w_e,\ F\subseteq\hat{E} \tag{7}$$

$$\delta(S)=\{\{i,j\}\in\hat{E}:\ i\in S,\ j\notin S\},\ S\subseteq N\cup\{0\}. \tag{8}$$

連結成分発見的解法では，デポを除去した後に支持グラフの成分を一つずつ考慮する．いま $S$ は，この

ような連結成分の頂点集合であるとしよう．$\hat{x}$ が $S$ によって決定される容量限定制約を満たさないときは，CUT 操作を実行する．そうでない場合には，除去後に制約が満たされなくなりそうな頂点 $v \in S$ を用いて $S \leftarrow S \backslash \{v\}$ を実行する（特に $b(C_i \backslash \{v\}) = b(C_i)$ と $w(\delta(C_i \backslash \{v\})) - 2b(C_i \backslash \{v\}) < w(\delta(C_i)) - 2b(C_i))$ を満たすような頂点 $v \in C_i$ を利用する）．このような手順は，頂点が見つからなくなるまで繰り返し実行される．不等号条件がすべて満たされたとすると，支持グラフの次の連結成分について考慮する．この方法の一つの変形版として2連結成分発見的解法があるが，この解法ではデポの除去後に2-枝連結成分（少なくとも2本の枝を除去して初めて非連結となるようなグラフ成分）から手順を開始する．

縮退型発見的解法では，デポに隣接していない支持グラフの任意の枝の二つの頂点が $\hat{x}$ によって容量制約が満たされない枝（すなわち，$S=\{i, j\}$ が容量制約を満たさないと仮定しよう）に対応するとき，CUT 操作を実行する．そうでない場合は，デポに隣接していない枝 $e=\{i, j\}$ が $w_e \geq 1$ を満たすとしよう．この場合，両端点を重ね合わせることによって枝 $e$ を縮退，あるいは縮約し，その結果得られた枝に対応する $w$ の成分を加え合わせる．このようにして得られた縮約グラフにおいて，同一化された二つの端点は $N$ の部分集合からなる超頂点を形成することになる．この新たな超頂点に対応する需要は，それに含まれる各頂点の需要の総和となる．この縮退操作が容量制約を満たさなくなる操作と矛盾しないことは容易に証明できる．すなわち $S$ が容量制約を満たさない場合，$i, j \in S'$ または $i, j \notin S'$ のいずれかを満たす集合 $S'$ が存在しなければならないことを示せばよい．このようにして，発見的解法では，$w_e \geq 1$ を満たす枝 $e$ を縮退させる操作と両端の頂点が容量制約を満たさないかどうかを調べる操作を交互に反復的に実行する．制約が満たされない場合は CUT 操作を実行し，そうでないときは，すべての枝 $e$ がデポに隣接しているか，あるいは $w_e<1$ を満たすようになるまで反復操作を繰り返す．縮退グラフにおいては，ある枝 $e$ に対して，$w_e>1$ となることが可能であることに注意されたい．

修正縮退型発見的解法は Nagamochi-Ibaraki[17] による最小カットアルゴリズムに基づいており，基本的には縮退型発見的解法に類似している．この修正版では，縮退操作は文献[17]のアルゴリズムに定められた順番に従って1より重みの小さい枝に対して実行される．縮退型発見的解法の場合と同様にして，それぞれの枝は容量制約を満たさなくなるかどうかを決定するために縮約する前にチェックされる．枝が選ばれる順番は各カットの重みが“小さい”順になっているので，このアルゴリズムでは，他の発見的解法では発見されないような容量制約を見出すことが可能となる．

上記の発見的解法が，容量制約を満たさないものを見出せない場合には，貪欲縮退型発見的解法を適用すればよい．この解法では，任意に選択されるか，あるいはまた何らかの発見的規則に従って選択されるような，小さな頂点集合 $S$ から出発する．それぞれの反復において，集合 $S$ に $\sum_{e \in \{\{i,j\} \in E: i \in S\}} x_e$ が最大（したがって $\sum_{e \in \delta(S)} x_e$ が最小）となるような頂点 $j \notin S$ を追加することによって，$S$ を大きくしていく．この戦略と，大きな初期集合が同様の規則を用いて縮退されるという連結成分発見的解法に基づく戦略とを対照させて見てほしい．

### 2.2 分割アルゴリズム

上述の発見的解法が容量制約を満たさないものを見出すことができなかった場合，文献[18]に最初に提示された分割アルゴリズムを適用する．ある LP 緩和問題に対する小数解 $\hat{x}$ が与えられたとき，$\hat{x}$ をツアの付随ベクトルの凸包として表すことによって $\hat{x}$ が多面体 $\Gamma$ の内部に存在するか否かを決定する．より厳密には，列ベクトルが $\Gamma$ の端点を表すような行列 $T$ に対して

$$\max\{0^T\lambda:\ T\lambda=\hat{x},\ 1^T\lambda=1,\ \lambda \geq 0\} \tag{9}$$

が有限な最適解を有するか否かを求めることが必要となる．$\hat{x}$ をこのようなツアベクトルの凸結合として分割可能な場合には，Carathéodory の定理によって $T$ の列の一部分のみが必要とされることが分かる．言うまでもないことであるが，行列 $T$ を作成することは容易ではなく，どのようにしてこれを実行するかという問題については，読者はもう少し待ってほしい．

$\hat{x}$ がツアベクトルの凸結合でありながら，容量制約を満たさない場合には，ある種の分割もまた同じ制約を満たさないことになる．このようにして，$\hat{x}$ に対する凸結合が与えられたとき，容量制約を満たさないようなものを求めるためには，ツアの中のデポからデポへのそれぞれの部分を調べることになる．この手続きが成功すると，満たされなかった不等式に対する CUT を作成する．この手続きがうまくいかない場合には，すなわち $\hat{x}$（したがって $\hat{x} \in \Gamma$）の凸分割が存

在して，さらに制約を満たさないものが存在しない場合には，分離が失敗したことになるので，BRANCH 操作を実行しなければならない．

$\hat{x}$ の凸分割が存在しない場合には，$\hat{x}\notin\Gamma$ となり，ファルカスの定理によって $\hat{x}$ を $\Gamma$ から分離する超平面が得られる．特に，$T$ の各行 $t$ に対して $at\geq\alpha$ であって，しかも $a\hat{x}<\alpha$ となるようなベクトル $a$ とスカラー $\alpha$ が存在しなければならない．この不等式は式(9)に定義されたLP問題を解く際の基底逆行列に対応するので，ただちに得られる．

行列 $T$ の処理に関しては2種類の方法が存在する．一つは行列 $T$ の列集合を限定することである．$\hat{x}$ の凸結合であるツア $t$ はすべて $\hat{x}$ に適合していなければならない．すなわち $t$ は，$\hat{x}_e=1$ のときはいつも $t_e=1$，$\hat{x}_e=0$ のときは $t_e=0$ を満たさなければならない．このようにして，これらの規定が $T$ のすべての列に対して成立することが必要となる．$\hat{x}$ の分数支持部分である，$0<\hat{x}_e<1$ を満たすような枝集合は要素数が少ないので，深さ優先探索法を用いて $T$ を明示的に生成することによってすべてのツアを数え上げることが可能となる．この方法の短所は，ファルカスの不等式がすべて“取り出され”なければならず，しかもそれは計算上かなり困難であるということである．

もう一つのアプローチは，列生成法を用いて $T$ を暗示的に処理するということである．ここで $T$ は最初はツアの部分集合族のみからなるため，アルゴリズムは前と同様に進行し，$\hat{x}$ が $T$ の列の凸結合であるか否かを求めることになる．凸表現が $T$ の列の中に求められた場合，ツアの一部分が容量制約を満たさないかどうかを調べる．容量制約を満たさない場合にはCUT操作を実行する．前と同様にして，既知のツアを用いて $\hat{x}$ の凸結合が存在しない場合には，ファルカスの定理によって，$T$ の各列 $t$ に対して $at\geq\alpha$ であって，しかも $a\hat{x}<\alpha$ となる組 $(a,\alpha)$ が得られる．ここでTSP最適化手法を用いてコストベクトル $a$ に関して $\Gamma$ 上で最小化操作を行う．$t^*$ が最適ツアの付随ベクトルであるとしよう．このとき，$at^*>\alpha$ ならば，最小化操作によって $ax\geq\alpha$ が $\hat{x}$ を $\Gamma$ から分離するため，この不等式がCUT操作に用いられる．また $at^*<\alpha$ の場合には，$t^*$ は $T$ の列には存在しないため，$t^*$ が $T$ に追加され，$\hat{x}$ の凸分割が $T$ の列から得られるか否かを再び求めることになる．この手順は分割が見つかるか，または $\hat{x}\notin\Gamma$ が証明されるまで繰り返される．

### 2.3 拡張

文献[11]に示された分割アルゴリズムの効率性の改善方法について述べる．$T$ の列を多面体 $P'$ の端点のみに限定する効果について考えてみよう．まず最初に，分割を見出すことができない場合には，われわれの目的が達成されないように思えるかも知れない．しかしながら，われわれが分割を見出せない場合に生成されたファルカスのカットを用いることによって $\hat{x}$ を $P'$ から分離することができ，さらにCUT操作としても利用することができる．この基本的なアプローチは，文献[2]にあるTSPに対する妥当な不等式を生成する場合にも利用することができる．

次にコストが現在の上界値に等しいか，それより小さい場合の解の付随ベクトルの凸包として定義される多面体 $P''$ について考えてみよう．$P''\subseteq P'$ はただちに得られるが，さらに $\min\{cx: x\in P'\}=\min\{cx: x\in P''\}$ も容易に得られる．このようにして，われわれは数え上げ操作をコストが現在の上界値より小さいような列のみに限定した場合にも，妥当なファルカスの不等式を生成することができる．

$T$ が空集合になる場合に，数え上げが点に限定されるとどうなるかを考えてみよう．この場合，アルゴリズムによって，コストが現在の上界値より小さいときには現在の分数解と適合するような実行可能解が存在しないことが証明される．したがってこの場合には，次のような非列カット（no-columns cut）を導入する．

$$\sum_{e:\hat{x}_e=1} x_e-\sum_{e:\hat{x}_e=0} x_e\leq|E_1|-1. \tag{10}$$

これらのカットは文献[5]にある仮想ツア（hypo-tours）の特別なケースである．実行可能性とコストの両者に基づく列生成を行うと，常にこれらの不等式の一つを生成することができる．このことを確かめるには，$T$ が多面体 $P''$ の端点のみからなるとしよう．このとき，これらの各列によって新たな上界値を得ることができ，さらにそれを $T$ から削除し，$T$ を空集合にすることが可能となる．

## 3. 計算結果

分離ルーチンはRalphs[18]，Ladányi[12]らによる遺伝的並行分岐カットによって構成されたSYMPHONYの中に組み込まれている．この方法の実施結果については文献[13]に示されている．われわれのテスト集合はTSPLIB[20]から取られた中規模の

VRP 問題と Augerat[3]による問題からなる．この論文に用いられた完全なテスト集合とソースプログラムのコードは http://www.branchandcut.org/VRP から得られる．

この集合にある 10 個の問題は文献[8]の Christofides と Eilon によって用いられたもので TSPLIB にある VRP 問題例にも含まれている．また 3 題（E-n 76-k 7，E-n 76-k 8，E-n 101-k 8）は 80 台のプロセッサによる IBM SP 2 並列コンピュータ上で SYMPHONY の並列版を用いて最初に解かれたものである．最近 Blasum と Hochstättler は追加的カット平面法[6]を用いて単一プロセッサ上で E-n 76-k 7 and E-n 76-k 8 を解いた．われわれも E-n 76-k 7 を直列的に解いたが，より大きなツリーを必要とした．われわれが知っている最小の問題例で未解決の問題が B-n 50-k 8 であることは重要であろう．われわれはトラック容量が 150 に増加した場合のこのモデルを解いたが，原問題については未解決である．

## 参考文献


[1] J. R. Araque, L. Hall, and T. Magnanti: Capacitated Trees, Capacitated Routing and Associated Polyhedra, Discussion paper 9061, CORE, Louvain La Nueve, 1990.

[2] D. Applegate, R. Bixby, V. Chvâtal, and W. Cook: TSP Cuts Which Do Not Conform to the Template Paradigm, *Computational Combinatorial Optimization,* D. Naddef and M. Jünger, eds., Springer, Berlin, pp. 261-303, 2001.

[3] P. Augerat: VRP problem instances, Available at http://www.branchandcut.org/VRP/data/, 1995.

[4] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberân, and D. Naddef: Separating Capacity Constraints in the CVRP Using Tabu Search, *European Journal of Operations Research*, 106, pp. 546-557, 1998.

[5] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberân, D. Naddef, and G. Rinaldi: Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem, Research Report 949-M, Universitē Joseph Fourier, Grenoble, France, 1995.

[6] U. Blasum and W. Hochstättler: Application of the Branch and Cut Method to the Vehicle Routing Problem, Zentrum für Angewandte Informatik, Köln, Technical Report zpr 2000-386, 2000.

[7] V. Campos, A. Corberân, and E. Mota: Polyhedral Results for a Vehicle Routing Problem, *European Journal of Operations Research*, 52, pp. 75-856, 1991.

[8] N. Christödes and S. Eilon: An Algorithm for the Vehicle Dispatching Problem, *Operational Research Quarterly*, 20, pp. 309-318, 1969.

[9] G. Cornuéjols and F. Harche: Polyhedral Study of the Capacitated Vehicle Routing Problem, *Mathematical Programming*, 60, pp. 21-52, 1993.

[10] G. B. Dantzig and R. H. Ramser: The Truck Dispatching Problem, *Management Science*, 6, pp. 80-91, 1959.

[11] L. Kopman: A New Generic Separation Routine and Its Application in a Branch and Cut Algorithm for the Vehicle Routing Problem, Ph. D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA, 1999.

[12] L. Ladânyi: Parallel Branch and Cut and Its Application to the Traveling Salesman Problem, Ph. D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA, 1996.

[13] L. Ladânyi, T. K. Ralphs, and L. E. Trotter: Branch, Cut, and Price: Sequential and Parallel, Computational Combinatorial Optimization, D. Naddef and M. Jünger, eds., Springer, Berlin, pp. 223-260, 2001.

[14] G. Laporte and Y. Nobert: Comb. Inequalities for the Vehicle Routing Problem, *Methods of Operations Research*, 51, pp. 271-276, 1981.

[15] G. Laporte, Y. Nobert, and M. Desrochers: Optimal Routing with Capacity and Distance Restrictions, *Operations Research*, 33, pp. 1050-1073, 1985.

[16] A. N. Letchford, R. W. Eglese, and J. Lysgaard: Multistars, Partial Multistars and the Capacitated Vehicle Routing Problem, Technical Report available at http://www.lancs.ac.uk/staff/letchfoa/pubs.htm, 2001.

[17] H. Nagamochi and T. Ibaraki: Computing Edge Connectivity in Multigraphs and Capacitated Graphs, *SIAM Journal of Discrete Mathematics*, 5, pp. 54-66, 1992.

[18] T. K. Ralphs: Parallel Branch and Cut for Vehicle Routing, Ph. D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA, 1995.

[19] T. K. Ralphs: SYMPHONY Version 2.8 User's Guide, Available at http://www.branchandcut.org/SYMPHONY, 2001.

[20] G. Reinelt: TSPLIB, A traveling salesman problem library, *ORSA Journal on Computing*, 3, 376-384. Update available at http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/, 1991.