解説

コンピュータサイエンスによる物流トラックの温室効果ガス排出削減

佐藤 一郎
国立情報学研究所

コンピュータサイエンスを直接的に利用して物流トラックにおける温室効果ガスの排出を削減する方法を示す．物流とコンピュータサイエンスは関係ないようにみえるが，トラックの輸送経路とプログラムの実行フローには類似性がある．そこでトラック経路を表すプログラミング言語を設計して，トラック経路をプログラムとして扱えるようにして，プログラムのための各種技術を通じてトラック経路を効率化できるようにする．本稿ではその中でもプログラム検証技術を利用して，共同物流管理を行う方法とコンパイラで利用されるコード最適化を用いて，トラック経路の効率化を行う方法を示していく．

はじめに

二酸化炭素 (CO2) を含む温室効果ガス（Greenhouse Gas）の排出を削減することは世界的な課題になっている．1997 年に京都で開かれた気候変動枠組条約第 3 回締約国会議 (COP3) において京都議定書が採択され，その中で日本は 2008 年から 2012 年の間に 1990 年比で約 6%の CO2 排出削減義務が課せられた．2009 年 12 月にコペンハーゲンで開かれた気候変動枠組条約第 15 回締約国会議 (COP15) では，国際的な削減目標は採択できなかったが，2009 年 9 月，鳩山首相は国連で開かれた気候変動首脳会合において日本は 2020 年には 1990 年比で 25%の CO2 排出を削減すると明言した．

しかしながら，CO2 排出削減は進んでいるとはいえない．すでに京都議定書の削減期間（2008 年から 2012 年まで）に入っているが，日本の CO2 排出量は減るどころか増えているのが現状であり，これは既存手法では限界があることを示している．まして 25%の CO2 排出削減を実現するには従来手法以外に，新しい手法が必要となる．

こうした状況において CO2 を含む温室効果ガスの排出手法として期待されているのが情報技術 (IT) の利活用である．たとえば総務省は CO2 排出削減，省エネルギー化に貢献する情報通信技術を支援するために地球温暖化対策 ICT イノベーション推進事業 (PREDICT) を進めている．

さて IT による現実世界の CO2 排出削減方法として，(1) 工場やビルの情報システムによるエネルギー管理の効率化を通じて CO2 排出削減する方法や，(2) 通信ネットワークにより物理世界の移動を減らして CO2 排出削減する方法などが提案されている．たとえば (1) は情報システムにより工場やビルの機械や空調を効率的に運用することで，エネルギーやリソースあたりの生産性をあげたり，エネルギー消費を減らす方法である．そして (2) は電子会議やオンラインコンテンツ配信により人や物が移動することを減らす方法である．ただし，これらは CO2 排出削減を支援するために IT を間接的に利用しているのに過ぎない．

一方，本稿では IT，それもその基盤であるコンピュータサイエンスの手法をなるべく直接的に使って，CO2 排出を含む温室効果ガスの排出を削減する方法を考えていく．さてコンピュータサイエンスと温室効果ガスの排出削減は無関係に思われるかもしれない．しかし，温室効果ガスの排出削減は現実世界の活動の効率化と表裏一体であることが多い．一方，コンピュータサイエンスは，かつてのコンピュータが高価かつ低性能だったことから，長年にわたってプログラム実行を効率化する方法が数多く研究されてきた．これらのコンピュータサイエンスにおける効率化手法を現実世界に応用できれば，現実世界の効率化，つまり CO2 排出削減ができる．そして本稿の目的の 1 つはコンピュータサイエンスをコンピュータ以外にも応用できることを示すことにある．

トラック輸送と CO2 排出削減

コンピュータサイエンスの対象であるコンピュータと現実世界は違うが，その一方で現実世界のなかにもコンピュータサイエンスの手法が適用できる分野がある．その 1 つが物流である．さて現代の物流はトラック輸送が中心になっているが，そのトラック輸送は CO2 を含む温室効果ガスの CO2 排出源になっている．環境省の速報値によると 2008 年度の日本の CO2 排出量は 12 億

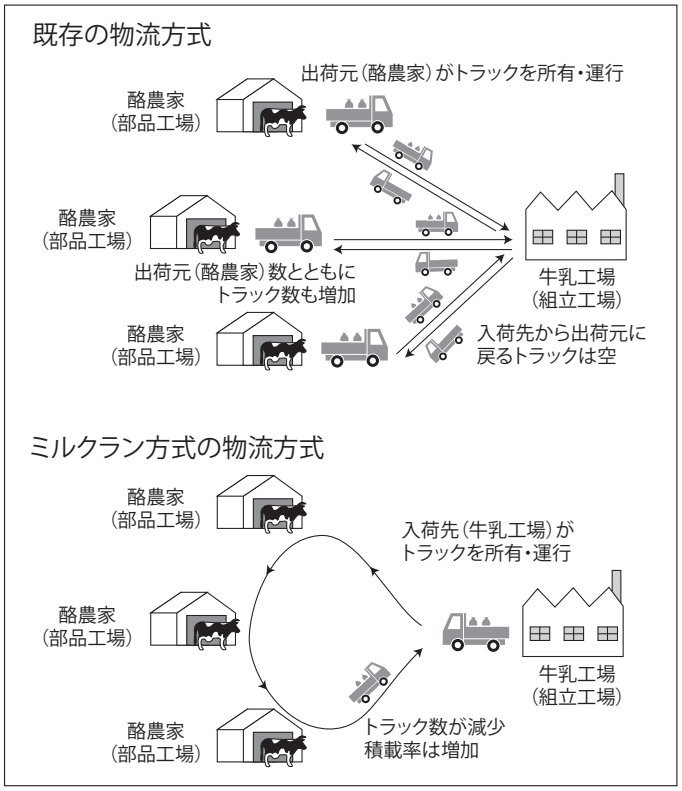

図 -1　共同物流の例(ミルクラン方式)

8,600 万トンである [1]．そのうち自動車・船舶などの運輸部門は 2 億 3,600 万トンであり，全体の約 2 割を占めており，その中でトラック輸送による CO2 排出量は多く，その削減は重要な課題となる．

これまでのトラック輸送における CO2 排出の削減方法としては，トラックのエンジンなどの低排出化や，GPS などを利用した道路運行管理，鉄道などの CO2 排出が少ない運輸手段と組み合わせるモーダルシフトなどが行われてきた．しかし，これらの局所的な削減手法では限界があり，特に 25%の CO2 排出削減には抜本的な方法が求められる．

そこで注目されているのが共同物流である．まず既存の物流では出荷元それぞれがトラックを用意して，入荷先に納品することが基本となっている．製造業，特に組立産業のように出荷元(部品工場)の数が入荷先(組立委)の数に対して多い場合はトラック台数が多くなる．さらに出荷元の荷物だけ運ぶため積載率も低く，入荷先から出荷元に戻るときはトラックの荷台は空であるなど，運用方法そのものが効率がいいとはいえない．さらに近年はジャストインタイム要求により，指定時間内の納品が求められるため，便数が増える一方で積載率は下がっている．

一方，共同物流にはいくつかの形態があるが，その 1 つは多数の出荷元が少数のトラックを共同で利用するものである．これによって，トラックの積載率をあげると同時に，トラック数を減らしていく．また，ミルクラン(Milk-run)と呼ばれる共同物流では，入荷先がトラックを用意して，複数の出荷元を回って荷物を集める(**図 -1**)．この方法は牛乳メーカのトラックが酪農家を回って牛乳を集めていたことから名付けられた．これは出荷元が多数だが入荷先が少ない場合は有効であり，実際，海外における自動車産業などでも利用が始まっている．出荷元の数よりも入荷先の数が多い場合も共同物利用は有効であり，複数事業者がトラックを共同運行することにより，トラック数を減らせる．たとえばコンビニエンスストアは同業他社の店同士が隣接することが多いことから，複数のコンビニエンスストア会社が店に商品を納品するトラックを共同で運行することにより，トラック数を減らせるといわれている．ところで CO2 排出削減という観点からは，トラックの移動距離の最短化よりもトラック数の削減の方が効果的とされる．それはトラックの生産で排出される CO2 量が大きいためである☆1．

しかし，共同物流はトラック輸送における CO2 排出削減の決定打と言われながら普及していない [2]．たとえば近年も 2 つの大手ビール会社が一部地域でビールの共同配送を始めることが大きく報道されたが，数カ月で頓挫したといわれる．共同物流が普及しない原因は管理の難しさにある．たとえばトラックの集配先や荷量はビジネス上の重要情報になることに加えて，出荷元・入荷先事業者ごとに輸送条件があり，各出荷元がトラックを用意することで対処してきたが，トラックを共同運行すると個々の条件に合わせるのが難しくなる．共同物流は路線バスのように複数のトラックが多様な経路かつダイヤで運行されることから，仮に条件に合ったトラックがあってもそれを見つけるのも難しい．実際，物流ではジャストインタイムのような時間指定の集配はもちろん，複数個所に荷物を届けることや，集配先で荷物を受け取ることがあるなどその条件が複雑であり，鉄道路線・ダイヤの検索手法では対応しきれない☆2．一方で共同物流による効果をあげるにはある程度の規模が必要であるが，これまでの共同物流では規模も小さく，当事者同士の属人的な調整を通じて管理していたが，大規模に共同物流を実現するには従来にはない方法が必要となる．

ところで，トラック輸送というと読者は宅配便を思い浮かべるかもしれない．しかし，宅配便トラックのように経路を動的に変更する運送は少数である．既存のトラック輸送の大部分は静的な経路，つまり輸送開始前に決められた集配個所を決められた順番通りにまわってい

☆1　トラック数の削減はトラック購入費や人件費の削減につながることから，経営的にも効果が大きい．
☆2　本稿では扱わないが，たとえばトラックに荷物を積載する際に荷量や荷姿，積載順序，温度などのさまざまな輸送上の条件がある．

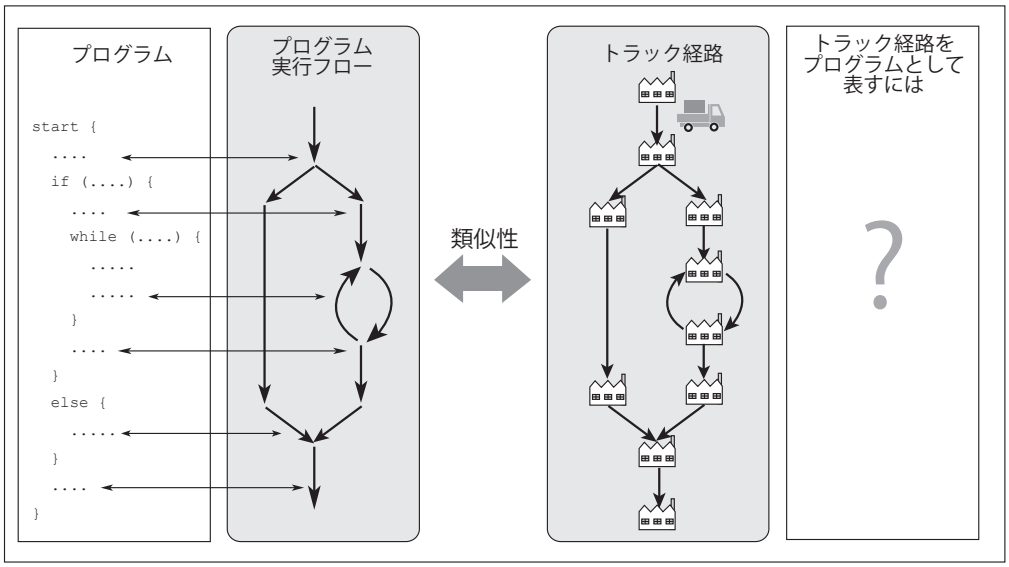

図 -2　プログラムの実行フローとトラック経路の類似性

る．このため本研究は静的な経路を対象とし，輸送中の経路変更は想定しない．

トラック経路記述言語

さて本稿ではコンピュータサイエンス，そのなかでもソフトウェア検証とコード最適化手法を利用してトラック輸送を管理・効率化する方法を考えるが，その基本アイディアの発案は筆者が物流の業界団体から物流効率化の相談を受けたところに遡る．その際に実際のトラック輸送経路を見せていただいたが，そこで気がついたのが，トラック経路とプログラムの実行フローの類似性である．**図 -2** のようにトラック輸送では複数集配個所のあいだを何度も往復することがある．これはプログラムでいうとループに相当する．また，トラックは複数集配個所のうち一部だけで集配すればいい場合があるが，これはプログラムでいうと条件分岐に相当する．そこでトラック輸送経路を表すためのプログラミング言語を定義して，トラック輸送経路をプログラムとして扱えるようにする．下記にその言語の定義(一部)を示す．

$$
\begin{aligned}
E ::= {} & \ell[t_1, t_2] \quad \text{(移動先)} \\
| \quad & E_1 ; E_2 \quad \text{(移動順序)} \\
| \quad & E_1 + E_2 \quad \text{(選択移動)}
\end{aligned}
$$

これはトラック経路特有の制御構造や到着時間を表せる専用言語であり，トラック経路をプログラムとして表せるようにする．なお，本稿はプログラムのための各種方法でトラック輸送を管理・効率化できることを示すことが目的であり，プログラミング言語などの詳細は省略する．文法やセマンティクスの定義はたとえば別稿[4),5)]を参照されたい．

簡単に言語を紹介すると，たとえば $\ell[t_1, t_2]$ は ℓ は集配先の名称または識別子であり，t_1 と t_2 は時刻であり，トラックは集配先 ℓ に時刻 t_1 から時刻 t_2 までに到着することを表す．そして演算子 ; はトラックの移動順序を表す[☆3]．演算子 + は選択を表す．たとえば**図 -3** のような経路は a[9:30, 9:40];b[10:00, 10:10];(c[10:20, 10:30];0+d[10:25, 10:35]);e[10:45, 11:00] として記述される．これはトラックは集配先 a に 9 時 30 分から 9 時 40 分までに到着し，次に集配先 b に 10 時から 10 時 10 分までに到着し，そして集配先 c または d のどちらか一方で集配を行うが，集配先 c に行く場合は 10 時 20 分から 10 時 30 分までに到着し，集配先 d に 10 時 25 分から 10 時 35 分までに到着する．そして最後に集配先 e に 10 時 45 分から 11 時までに到着することを表す．

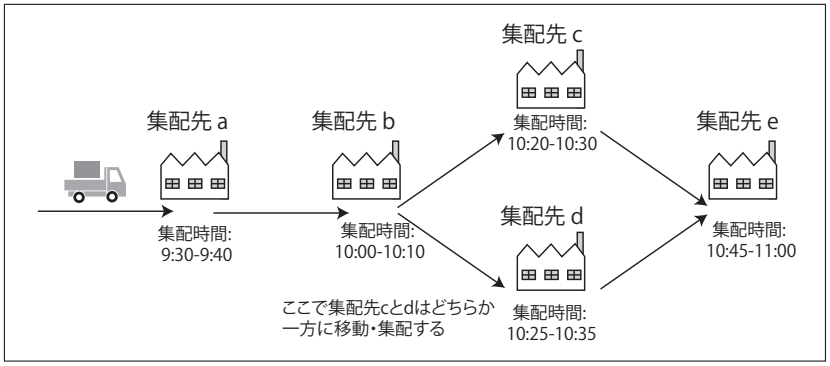

図 -3　トラック経路の記述例

なお，トラック輸送では積載される荷物の種類，荷姿，荷量が重要になる．本稿ではページ数の都合で荷物の記述は扱わないが，この言語ではトラックの積載スペースは変数として，そして荷物は変数に格納される定数として導入される．そして，荷物の種類および荷姿，荷量などはデータ型として導入される．トラックの積載では荷

☆3　ここで時刻は絶対時間となり，t_1 より t_2 は後の時刻とする．また $\ell_1\{t_1, t_2\};\ell_2\{t_3, t_4\}$ のように演算子 ; で結ばれた場合，t_1，t_2，t_3，t_4 は時間的な整合性がとれている，すなわち t_3 は t_1 よりも後の時刻，t_4 は t_2 よりも後の時刻とする．

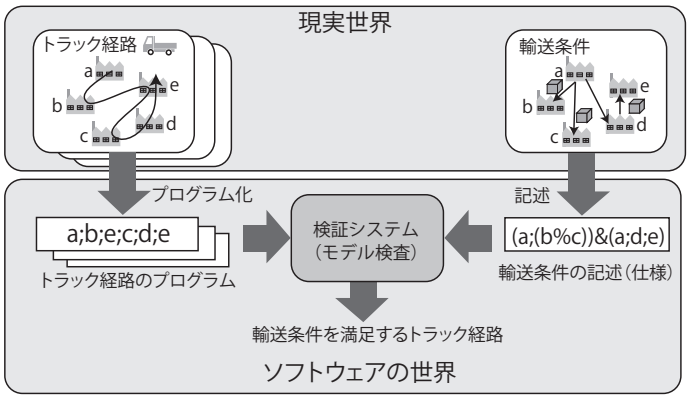

図 -4　検証技術を利用した輸送条件を満足するトラック経路の判定

物の形や大きさなどに制限されることがあるが，荷物の大きさや形に対応したデータ型に対する型推論を通じて積載可否が判定される．

前述のようにトラック経路をプログラムとして表せることから，プログラムのための各種手法がトラック経路に応用できることになる．まずソフトウェア検証技術を使った共同物流管理を示し，次にコード最適化を利用したトラック経路効率化を示す．両章の手法は組み合わせることもできるが，独立しても利用できる．

ソフトウェア検証によるトラック選択

共同物流では輸送の条件を満足するトラックを発見し，もし複数台のトラックが満足する場合は移動距離が短い，つまり CO2 の排出量の少ないトラックを選ぶことになる．ここではトラック経路をプログラムとして扱えることから，検証（Verfication）の手法を利用していく．つまりトラック経路を実装したプログラムとして扱うとともに，輸送の条件をプログラムの仕様としてみる．これによりトラックが輸送の条件を満足するか否かは，プログラムが仕様を満足するか否かを検証することに帰着する（**図 -4**）．そこでまず輸送の条件の記述言語を考える．出荷元または入荷先からみるとトラックの集配先とその集配時刻が重要となることから，移動順序と時刻に特化した記述言語を定義していく．

$$
\begin{aligned}
E ::= {} & \ell[t_1, t_2] && \text{（集配先）}\\
| \ & E_1 >_T E_2 && \text{（集配順序）}\\
| \ & E_1 \# E_2 && \text{（選択的集配）}\\
| \ & E_1 \,\%\, E_2 && \text{（可換的集配）}\\
| \ & E_1 \,\&\, E_2 && \text{（集配条件の合成）}
\end{aligned}
$$

ここで $\ell\{t_1, t_2\}$ は集配先 ℓ に t_1 時間から t_2 時間に集配することを求めている．また $>_T$ は集配順序を表し，T は最長輸送時間である[☆4]．食料品や原材料などの輸送などではトラックに積載されている時間に制限がある場合が多く，その場合にだけ T にその時間を指定する．たとえば $a > b$ は集配先 a で集配後にいずれは集配先 b で集配することを求めている．記号 # は選択を表し，たとえば $a \# b$ は集配先 a と b のどちらか一方に移動・集配をすればいいことを表す．そして，記号 % は集配順序が不定であることを表し，たとえば $a \,\%\, b$ は集配先 a と b の両方で集配する必要があるが，その集配順序はどちらが先でも構わないことを表す．記号 & は複数の輸送条件の合成であり，複数の輸送条件を列挙する場合に用いる．

ここで共同運行されているトラックが 3 台あり，その経路は下記とする．

トラック 1　b[9:10, 9:20];a[10:10, 10:20]
トラック 2　a[9:30, 9:40];b[10:40, 10:50];
　　c[11:10, 11:20]
トラック 3　a[9:00, 9:10];c[9:30, 9:40];
　　b[10:10, 10:20]

輸送条件 $a\{9{:}00, 9{:}30\} > b\{10{:}00, 10{:}30\}$ を考える．これはたとえば集配先 a で 9 時から 9 時 30 分までに荷物を積み込み，集配先 b に 10 時から 10 時 30 分までに届けることを表す．この条件を満足するトラックだが，トラック 1 は集配順序が条件に合わない．トラック 2 は集配順序そのものは条件に合致するが，集配時刻が条件に合わない．トラック 3 は途中で集配先 c に立ち寄るものの集配順序・時刻とも条件に合致する．

次に輸送条件 $a\{9{:}30, 10{:}00\} > (b\{9{:}30, 11{:}30\} \,\%\, c\{9{:}30, 11{:}30\})$ を考えてみる．これはたとえば集配先 a で 9 時 30 分から 10 時までに荷物を積み込み，集配先 b と c の両方に集配順序は問わないが，9 時 30 分から 10 時 30 分までに届けることを表す．トラック 2 と 3 は条件に合致する．輸送時間に制限を加えた輸送条件 $a\{9{:}30, 10{:}00\} >_{1:30} (b\{9{:}30, 11{:}30\} \,\%\, c\{9{:}30, 11{:}30\})$ の場合はトラック 3 は条件を満足するが，トラック 2 は輸送時間が輸送時間の制限 (1 時間 30 分間) を超えてしまう．

さてトラック経路が輸送条件に合致するかの判定は，ソフトウェア検証で利用されるモデル検査（Model-Checking）手法を利用している．これは $>_T$ は時相論理における未来を表す演算子（線形時相論ならば◇演算子に，分岐時相論理ならば $\mathcal{F}$ 演算子）と同様なためである．そして各トラックの経路をグラフ構造に展開して，そのグラフ構造上で輸送条件が満足するかを判定している．複数トラックが条件を合致したときは CO2 排出量の少

☆4　たとえば輸送条件 $\ell_1\{t_1, t_2\} >_T \ell_2\{t_3, t_4\}$ において，t_1，t_2，t_3，t_4，T は時間的な整合性がとれているとする．

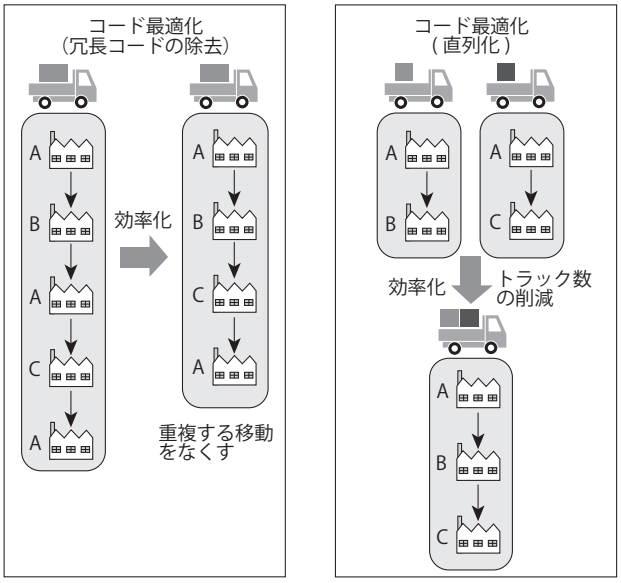

図 -5　コード最適化を利用したトラック経路効率化の一例

ないトラックを選択する．なお，トラックの CO2 排出量は基本的にはトラック移動距離に比例していることから，条件が合致するトラック経路の集配先間距離の和が小さいものを選ぶ．このとき各集配先間のトラック移動距離は既知とする．なお，モデル検査以外による判定も可能であり，プロセス計算などで用いられる双模倣性を利用した方法は別稿[5]に示されている．

経路効率化とコード最適化

トラック経路をプログラムとして表せることにより，プログラムのための効率化手法がトラック輸送に利用できるようになる．その手法の 1 つがコンパイルにおけるコード最適化である．その一例を紹介する．代表的なコード最適化手法に冗長コード除去がある．これはプログラム上ですでに計算されている結果を再利用することで，同じ計算を再度行う無駄を省く方法だが，これを利用することによりトラック輸送において同じ集配先に何度も行かないように経路を変更することになる（**図 -5**）．また，トラック輸送の CO2 排出量を削減するにはトラックの数を減らすことが求められるが，複数トラックによる集配経路は並列プログラムに相当する．一方，並列プログラムなどでは逐次実行プログラムの並列化とは逆に並列プログラムを逐次化(直列化)することがある．この逐次化を複数トラックの集配経路に適応することで，集配自体は変更せずにトラック台数を減らすことができる．筆者が実際のトラック輸送経路に対して，所定の輸送条件を満足させながら，コード最適化手法により効率化したところ，10％程度の効率化ができることが確認できている．これは比較的効率化されているトラック経路における結果だが，実際のトラック輸送ではその経路には無駄が多いことから，効果はさらに大きくなると予想される．

ここで留意しないといけないのは現場のニーズとの合致である．トラックによる CO2 排出量を減らすには移動距離を短くすることになるが，移動距離の短い経路を発見する方法として巡回セールスマン（Traveling Salesman）問題が知られている．巡回セールスマン問題はアルゴリズム効率化など学術的には重要な課題かもしれないが，実際のトラック輸送では利用されているとは言い難い．その理由は現実のトラック経路は距離だけでなく，渋滞や信号数も考慮して決められることも多いことに加えて，現場のニーズとの深刻なミスマッチがある．トラックを運転するドライバーは走り馴れた道を好むため，大きな経路変更は受け入れない傾向がある．一方，巡回セールスマン問題を想定した既存アルゴリズムは全体最適化手法であるため，最短経路を発見しても既存経路と大きく異なることが多い．つまり机上では有効な方法も現場の制約を無視していると利用されないことがある．なお，本稿で示したコード最適化は既存経路に対する部分最適化手法であり，トラック事業者への負担は小さい．

実証実験に向けて

さて本稿で示した共同物流管理や経路効率化は，総務省の地球温暖化対策 ICT イノベーション推進事業に採択され，凸版印刷（株）と日本ユニシス（株）とともに 2010 年後半と 2011 年前半に実際の物流を用いた実証実験を行う予定であり，現在のその準備が進められている．

大規模な共同物流を想定していることから，共同物流管理は**図 -6** のように PaaS（Platform as a Service）型クラウドコンピューティングインフラである Google App Engine 上に実装されている．トラック経路を格納するデータベース，そして各トラック経路が荷主の輸送条件を満足するかを調べる充足判定エンジンからなる．トラック事業者はトラックを運行する前にその経路を表すプログラムをデータベースに登録する．荷主は記述言語で表現された輸送条件とともに問合せを行う．充足判定エンジンは輸送条件を受け付けるとデータベースで登録された経路の中で条件と合致するものを探す．複数トラックが条件に合致したときは移動距離の短い，つまり CO2 排出の少ないトラックの選択して，荷主に推奨する．

プロトタイプ実装ではトラック数が 100 台程度であれば 1 秒以内に判定できている．これは多くのトラック輸送では 1 回の運行でトラックがまわる集配個所の数は 10 カ所を超えることは希なため，クラウドコンピュー

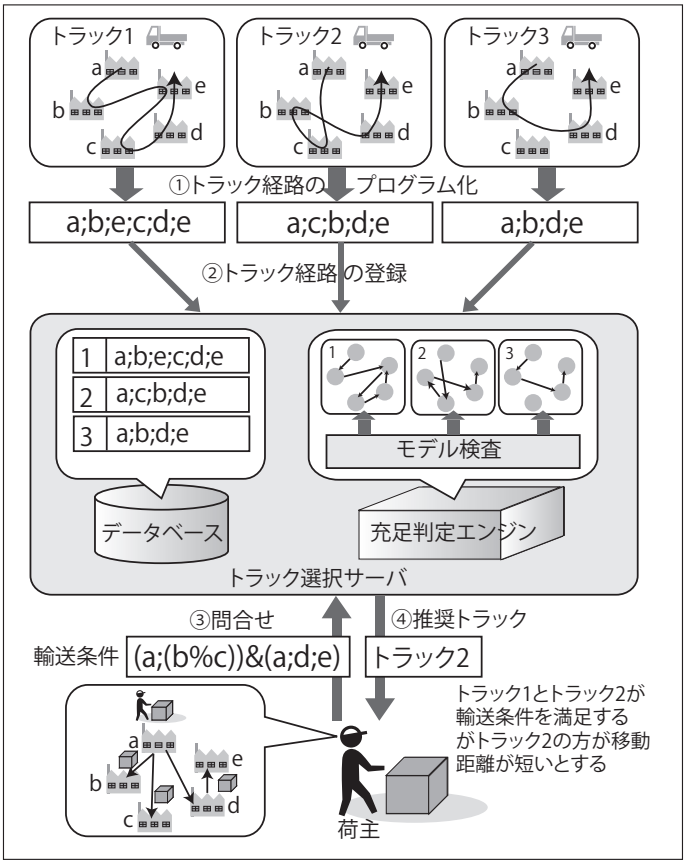

図 -6　トラック選択システムの構成

ティングのように並列度が上げられる環境では判定コストが小さくなるためである．

ところでトラックの経路は記述言語によるプログラムとして登録の必要があるが，同言語は運行計画表との親和性を考慮しており，運行計画表から機械変換も可能と思われる．また，輸送の条件についても GUI などにより簡易に入力する方法を検討している．このためトラック事業者も荷主側もプログラムを記述しなくも済むようにする．なお，トラックへの改造も不要であり，トラック事業者への負担は小さい．

おわりに

これまでのコンピュータサイエンスの研究は，コンピュータにおける原理解明に重点が置かれてきた．しかし，IT が現実世界で広く使われ，現実世界から IT に対する期待も大きくなっている現状では，コンピュータの原理解明だけでなく，現実世界の問題解決に貢献することも重要である．一方でコンピュータサイエンスのなかでも本質的な原理ならば適用分野を選ばないはずである．逆にそれができなければコンピュータサイエンスはコンピュータという狭い世界に閉じた学問分野になってしまう．

そこで本稿ではコンピュータサイエンスがコンピュータ以外にも利用できること示すために，コンピュータサイエンス，そのなかでもソフトウェア検証とコード最適化を利用することで，トラック輸送における CO2 排出を削減する方法を示した．

なお，本稿はソフトウェア検証とコード最適化を現実世界に応用したが，これ以外にもコンピュータサイエンスの多くの成果が現実世界においても有用と思われる．たとえばインターネットをはじめとするパケット通信は初期において物流の仕組みが大きく影響しているが，その後，高度に発達したパケット通信技術を逆に物流に導入することで，物流の効率化に貢献できるはずである．実際，本稿で示したトラック経路の記述言語はネットワーク管理技術の研究で開発された言語がベースになっている [3]．また，オペレーティングシステムは計算リソースの共有などの有効利用が大きな目的となっているが，そのオペレーティングシステムにおけるスケジューリング技術を社会リソースの共有に利用することは不可能ではないはずである．今後のコンピュータという枠を超えた，コンピュータサイエンスの発展・応用に期待したい．

本稿で示したコンピュータサイエンスによる CO2 排出削減手法を広範囲に展開するために，多くの協力者・参加者が必要であり，研究に協力していただける研究者・技術者を募っている．また，実証実験に協力していただける企業で興味のある方は筆者に問い合わせていただきたい．

謝辞　本活動の一部は，総務省「地球温暖化対策 ICT イノベーション推進事業（PREDICT）」の支援により実施した．また，(財)日本ロジスティクスシステム協会をはじめに多くの物流業界の方々から物流に関するさまざまな知識をいただいた．

参考文献

1) 環境省 : 2008 年度 (平成 20 年度) の温室効果ガス排出量 (速報値)，環境省 (Nov. 2009).
2) 月刊ロジスティクス・ビジネス編集部 : 共同物流入門 : 成功と失敗を分けるもの，LOGI-BIZ, Vol.8, No.10, pp.14-15 (Oct. 2008).
3) Satoh, I. : Building and Selecting Mobile Agents for Network Management, Journal of Network and Systems Management, Vol.14, No.1, pp.147-169, Springer (Jan. 2006).
4) Satoh, I. : A Specification Framework for Earth-friendly Logistics, in Proceedings of 28th IFIP WG6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'2008), Lecture Notes in Computer Science (LNCS), Vol.5048, pp.251-266, Springer (June 2008).
5) Satoh, I. : A Formal Approach for Milk-run Transport Logistics, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E91-A, No.11, pp.3261-3268 (Nov. 2008).

(平成 21 年 12 月 8 日受付)

佐藤 一郎（正会員） ichiro@nii.ac.jp
1991 年慶應義塾大学理工学部電気工学科卒業．1996 年同大学理工学研究科計算機科学専攻後期博士課程修了．博士（工学）．2001 年国立情報学研究所助教授．2006 年より同研究所教授．総合研究大学院大学複合科学研究科情報学専攻教授（併任）．