

車両運用計画最適化への並列処理技術の適用

平方 敦† 小高 知宏‡
†福井大学工学部

黒岩 丈介‡ 白井 治彦†
‡福井大学工学研究科

1. はじめに

本研究では、遺伝的アルゴリズムおよび、Graphic Processing Units(GPU) による並列処理を用いて車両運用計画の自動導出を行う。現在、車両運用計画についてはPCを用いて自動導出する研究が行われており、人間の手によって作成された車両運用計画と比較しても遜色のないものを導出することができている [1]。しかし、列車の本数が多い路線の車両運用計画をPCを用いて作成しようとした場合に、処理時間が長くかかってしまうという問題点がある。多数の演算器を持ち並列計算可能なGPUは、近年、CPUをはるかに上回る演算性能を示している [2]。CPUが少数のコアで逐次行う演算を、GPUは多数のコアに分配して演算することが可能であるため、実行される処理によってはGPUの処理時間がCPUの処理時間より短くなりうると考えられる。よって本研究の目的は、遺伝的アルゴリズムを用いて車両運用計画をモデル化することで、環境適応度算出処理をGPUデバイス上で並列処理することにより、計算処理を効率化することである。

2. GAを用いた車両運用計画のモデル化

本研究では、遺伝的アルゴリズムを用いることで、車両運用計画のモデル化を行う。ここでモデル化を行う上で、車両運用計画について簡単に述べる。車両運用計画とは、鉄道車両を一日の中でどの列車に充当させるかを定める作業である。車両運用の効率は、鉄道事業者が消費するコストの大小に直結しており、高い品質での車両運用計画の導出は非常に重要な課題である。車両運用計画において、1日ごとに各車両に割り当てる仕業を変えていき、各車両が運用計画内の全仕業に循環的に割り当てられるような車両運用計画を考えた場合、使用される全ての車両は、均等な走行距離とすることができる。このような複数日にわたる車両の循環的な使用計画を「交番」という [3]。交番を満たす車両運用計画を実現するために、計画を立てる路線の列車運行時刻表から、各列車を頂点とし、前に充当された列車から次に充当される列車までの待ち時間をアークとしたネットワーク

モデルを作成する。全てのノードを一度ずつ通る巡回路に含まれるアークのコストが最小となるものを導くことで、交番を満たしつつ、待ち時間が最小となる車両運用計画を立てることができる。運用内の待ち時間が最小となる車両運用を求める際に、本研究では遺伝的アルゴリズムを用いる。ネットワークモデルにおける各ノードを遺伝子とし、ノードを巡回する順序を遺伝子配列とする。各運用*i*における運用内の待ち時間の合計値である環境適応度 C_i は以下の式で表される。

$$C_i = \sum_{j=1}^n (Ts_j - Te_{j-1}) \quad (1)$$

式(1)において、 Te_j は j 番目に割り当てられる列車の終着駅時刻であり、 Ts_j は j 番目に割り当てられる列車の始発駅時刻を表している。 n は列車の本数であり、環境適応度 C_i の値は小さいほど待ち時間が少ない車両運用計画を意味している。次に、親となる運用の選択には、ルーレット方式を用いた。各個体 i が親に選択される確率 P_i は、以下の式で表される。

$$P_i = \frac{C_{max} - C_i + C_{min}}{\sum_{j=0}^m (C_{max} - C_j + C_{min})} \quad (2)$$

式(2)において、 C_{max} は親の候補となる運用 C_i における環境適応度の最大値を表しており、 C_{min} が環境適応度の最小値を表している。また、 m は個体数を表す。親の選択では、各個体 i が親に選択される確率 P_i の値が大きいほど、親として選択される確率が高いことを示している。親に選出された運用同士を順序交叉を用いて交叉を行い、次の世代の運用を生成する。上述に示す選択、交叉、突然変異の操作を繰り返し行うことで、環境適応度 C_i がより少ない車両運用を導出することが可能となる。車両運用の環境適応度 C_i の計算には、GPUを用いた並列処理技術を適用するものとする。GPUの1つのスレッドあたり、1つの運用の環境適応度 C_i の計算を行った。遺伝子配列の要素数である n 個のスレッドを用いて並列処理するものとした。

実験においては、運用を作成する対象として、小規模路線の時刻表データと、標準的な規模の路線の時刻表データを用いた。小規模路線の時刻表データでは $n = 12$ とし、標準的な規模の路線の時刻表データでは $n = 68$ とした。時刻表データには、対象とする路線で運行している列車の始発駅の発時刻と終着駅の着時刻、また列車の固有番号が含まれるものとしている。今回の車両運用導

Railway rolling stock operation optimization using parallel processing

†Atsushi Hirakata ‡Tomohiro Odaka ‡Josuke Kuroiwa †Haruhiko Shirai

†Faculty of Engineering, University of Fukui

‡Graduate School of Engineering, University of Fukui

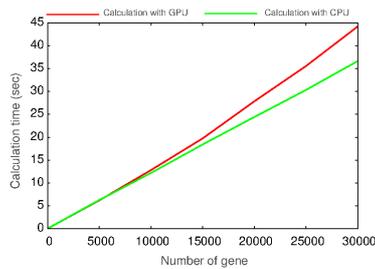


図 1: 小規模路線 $n=12$ での車両運用導出における処理時間

出においては,生成される解の集団の数 $m = 1280$ とし,その中から親の個体として 30 個の解を選出するものとした。

3. 車両運用の導出実験

小規模路線の時刻表データと,標準的な規模の路線の時刻表データに対して車両運用の導出実験を行った。時刻表ごとの車両運用の導出に要した時間を図 1, 図 2 に与える。実験では,0 世代から 30000 世代まで,5000 世代ごとに車両運用計画を導出した。図 1 は小規模路線の時刻表データに対して車両運用の導出実験を行った場合の処理時間の結果である。図 1 より,時刻表データに含まれる列車の数 m が少ない場合の車両運用計画の導出においては,1 世代から 15000 世代の間では両者の処理時間の差は殆どなく,概ね 1 秒以内となっている。しかし 15000 世代から徐々に実行時間の差が大きくなり,30000 世代計算を行った場合 CPU のほうが GPU よりも約 8 秒,高速に演算を行えている。図 2 は標準的な規模の路線の時刻表データに対して車両運用の導出実験を行った場合の処理時間の結果である。図 2 より,時刻表データに含まれる列車の本数 m が多い場合の車両運用計画の導出では,全ての世代数において CPU よりも GPU のほうが高速に演算処理を終えることが分かる。また,15000 世代から指数的に実行時間の差が大きくなり,30000 世代分計算を行った結果,CPU における実行時間と GPU における実行時間の差は約 40 秒であった。よって,小規模路線の車両運用の導出では GPU よりも CPU のほうが高速で演算を終え,標準的な規模の路線の車両運用の導出では CPU よりも GPU のほうが高速に演算を行う結果となった。

4. 考察

小規模路線の時刻表データでの車両運用導出において,CPU のほうが GPU よりも高速に演算したことについて考察する。車両運用のコスト計算を GPU 端末上で行う際には,CPU 上から計算に必要な配列データを GPU 端末上にコピーし,計算結果を GPU 端末上から

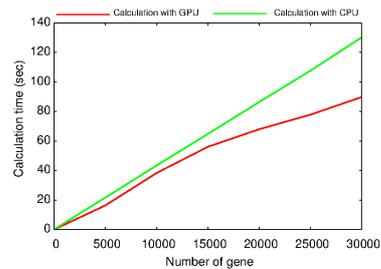


図 2: 標準的な規模の路線 $n=68$ での車両運用導出における処理時間

再度 CPU 端末にコピーする必要がある。この処理には非常に時間がかかり,世代数分繰り返した場合に,CPU を単独で用いる場合との実行時間の差が大きくなってしまふと考えられる。また,列車の本数が少ない場合の計算量については,車両運用のコスト計算を CPU 上で逐次行った場合と,GPU 上で並列的に行った場合で,計算量の差は比較的少ないことから,CPU のほうが GPU よりも高速に演算を終えたと考えられる。また,標準的な規模の路線については,列車の本数 m が比較的多く,計算量も増加することから,個体値の計算の逐次処理と並列処理の差が大きくなり,CPU より GPU のほうが高速で演算を行う結果となったと考えられる。

5. まとめ

本研究では車両運用計画を遺伝的アルゴリズムを用いて導出する際に,GPU を用いた並列処理技術の適用を行った。標準的な規模の時刻表データの車両運用導出において,GPU を用いることで,処理の高速化が実現できた。列車本数の多い路線については,考える車両運用の組み合わせの数についても膨大な数となり,人間の手による車両運用の計画は困難である場合が多い。こういった場合には,GPU 端末を使用した並列処理技術を用いて車両運用計画導出する手法は有効であるといえる。

参考文献

- [1] 福村直登 中村達也 西森進矢 坂口隆, 車両運用計画自動作成アルゴリズムの開発, 鉄道総研報告, Vol.22, No.6, pp.5-10, 2008.
- [2] 大磯正嗣 松村嘉之 保田俊行 大倉和博, CUDA 環境におけるデータ並列化を用いた遺伝的アルゴリズムの実装手法, 知能と情報 (日本知能情報フェジ学会誌), Vol23, No.1, pp.18-28, 2011.
- [3] (財) 鉄道総合技術研究所運転システム研究室編, 鉄道のスケジューリングアルゴリズム, エヌ・ティー・エス, 2005.