

International Conference on Computational Science, ICCS 2013

Solving Multi-criteria Vehicle Routing Problem by Parallel Tabu Search on GPU

Jagiełło Szymon^a, Żelazny Dominik^{a,*}

^a*Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland*

Abstract

Transportation plays a crucial role in both production and service industry. In modern times, the importance of supplying the goods on time to warehouses, production units and finally to the customers is not lost on logistic companies. Vehicle Routing Problems (VRP) models evolved to be more advanced, which led to the growth of computational complexity. Optimizing transportation routes for companies means performing complex computations and doing so in the shortest possible amounts of time. Graphics Processing Units (GPUs) provide massive computation when the needed operations are properly parallelized. nVidia GPUs are equipped with Compute Unified Device Architecture (CUDA), so applying parallel algorithms is not limited to complex workstations or specialized computers. This work emphasizes the value of using parallel Tabu Search (TS) algorithm over sequential TS algorithm and its application to multicriteria discrete optimization of Distance-constrained VRP.

Keywords: multi-criteria; parallel Tabu Search ; distance-constrained VRP ; OpenCL

1. Introduction

Vehicle Routing Problem, being the expansion of classical Traveling Salesman Problem (TSP), is a discrete optimization problem consisting of assigning certain number routes and keeping the constraints. First model of the problem was proposed in the late fifties/early sixties, when Dantzig and Ramser [1] developed mathematical programming formula and algorithm solving gas delivery to service stations. Since then, the vehicle routing evolved and is currently commonly studied. It is considered NP-hard problem of combinatorial optimization. Finding an optimal solution is possible only for small sized instances [2]. Heuristic algorithms won't guarantee finding the optimal solution, but in practice are most commonly used due to their high performance. The main purpose of the original problem was to minimize total length of vehicles routes. Solution of the classical VRP is a set of feasible, meeting the constraints, routes beginning and ending in starting point. Real world applications usually require more complex models, based on additional requirements and constraints like vehicle capacity, maximum route length or drivers worktime, pickup/delivery time windows and many others. There are also pickup and delivery models, which consider supply and collection of goods from clients.

In over a half century VRP evolved in relation with real world demands of the industry and logistics companies. In return many different problem types were formulated, like the ones with many starting points (depots), periodic VRP, VRP with split delivery, returns and many others. This work describes Tabu Search (TS) algorithm, which is able to find good solutions in certain computational time, proposed for Distance-constrained VRP.

*Corresponding author: Dominik Żelazny.

E-mail address: dominik.zelazny@pwr.wroc.pl.

2. Literature on multi-objective optimization and VRP problem

The literature on multi-objective optimization is in abundance, albeit the multi-criteria transportation problems have not received such interest. Especially in relation to the number of works on the same problems with single criterion.

2.1. Multi-objective transportation optimization

As mentioned before, over the last years, many techniques have been proposed for multi-objective optimization problems. They consist of scalar methods (ie. metacriteria), Pareto methods (ie. NSGA-II) and those that don't belong to either of the above (ie. VEGA). Some researchers approached the multi-objective transportation problem. A few of those approaches will be mentioned here.

One of the promising scalar techniques was proposed by Bowerman et al. in [3]. It uses five different sets of weights chosen by a decision-maker. An insertion algorithm was used by Lee and Ueng in [4]. In each iteration it adds one node to the vehicle with the shortest work time using a saving criterion. Another insertion heuristic was proposed by Zografos and Androutopoulos in [5], although its origin was in a method proposed by Solomon. It differs in the selection of the customers to be inserted, allowing both routed and unrouted demand points to be inserted. In [6] Pacheco and Marti optimize the makespan objective for every possible value of the second objective and then use a Tabu search algorithm to solve each problem. Similar strategy was used by Corberan et al. [7], but instead of Tabu search, they used scatter search approach.

In multi-objective vehicle routing problems, the Pareto concept is frequently used within an evolutionary framework. One of the works on uses a memetic algorithm proposed by Żelazny in [8]. This genetic algorithm in each iteration uses local search method on non-dominated solutions in order to further improve the Pareto frontier approximation. Pareto dominance has also been used by Ulungu et al. in a simulated annealing technique called Multi-Objective Simulated Annealing (MOSA) [9]. Also Paquete et al. [10] have called upon Pareto Local Search techniques.

Some studies employ neither scalar nor Pareto methods to solve multi-objective routing problems. These non-scalar and non-Pareto methods are based on lexicographic strategies or specific heuristics. Aforementioned VEGA [11] algorithm might be included, as an example of those specific heuristics. While lexicographic strategy was used in works of Keller and Goodchild [12] [13].

3. Problem description

Typically, VRP is described as follows. We are given a fleet of vehicles (usually identical) $V = \{1, \dots, v\}$, a set of customers/locations represented by nodes $N = \{1, \dots, n\}$, a starting node called depot (commonly referred to as node 0) and a network of connections between depot and customers/locations. For each pair of locations (i, j) , where $i, j \in (N \cup 0)$ and $i \neq j$, there is associated a route length $d_{i,j}$ or drive time $t_{i,j}$.

This paper describes Distance-constrained VRP problem, in which each vehicle is constrained by maximum route length it can travel. Meaning, that each vehicle has maximum distance a_k and feasible solutions is one, where vehicles can't exceed their maximum distances.

3.1. Criteria and model constraints

In our work, we used to objective functions: a) average route length and b) maximum route of a single vehicle. We based our work on Distance-constrained VRP model, so the maximum distance constraint was added to each vehicle.

First criterion, max route length per vehicle, is denoted as follows:

$$\max_{1 \leq k \leq m} \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}^k, \quad (1)$$

while second criterion, average rout length, is represented by following equation:

$$\frac{1}{m} \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ij} x_{ij}^k, \quad (2)$$

Where:

$x_{ij}^k = 1$, if j -th client is adjacent to and visited after i -th client, 0 otherwise;

d_{ij} = route length from i -th to j -th client.

The fleet of vehicles in our model is homogenous, so we consider the problem with maximum distance constraint $a_k = a$, where $k \in V$.

4. Multi-objective parallel Tabu Search algorithm

In our work we decided to use a modified Tabu Search algorithm, with external archive of non-dominated solutions. Our Parallel Pareto Archived Tabu Search (PPATS) was designed for neighborhood search of Pareto efficient solutions. Since multi-objective optimization with Pareto approach means finding an approximation of Pareto frontier, using a parallel local search method provides us with few design concepts unused in sequential algorithms due to their high computational complexity.

Tabu search, proposed by Glover [14], is a metaheuristic local search algorithm used for solving combinatorial optimization problems like production scheduling or, in our case, vehicle routing. It uses a neighborhood search procedure to move from one potential solution x to an improved solution x' in the neighborhood of x . In order to search space that would be left unexplored by other local search procedures, Tabu search carefully explores the neighborhood of each solution as the search progresses. The solutions admitted to the new neighborhood, $N(x)$, are determined through the use of Tabu list, a set of rules and banned solutions used to filter which solutions will be admitted to the neighborhood $N(x)$ to be explored by the search. We used its simplest form, a short-term set of the solutions that have been visited in the recent past.

In each iteration, the algorithm performs a number of parallel neighborhood searches, with differed starting points, for solutions from current Pareto frontier. During the search solutions are compared with their predecessors, if the new solution dominates the old one, it takes its place in next iteration of the search. If it is dominated by the predecessor, then it gets discarded and we continue with a current solution. Otherwise, if neither of the solutions dominates the other, then we add a new solution to the Pareto frontier and choose the one with lower normalized sum of objectives to take place in another iteration of the search. After each iteration of parallel neighborhood searches, the obtained solutions are checked by a selection function and non-dominated ones added (if they don't duplicate the existing solutions) to the external set and we purge this Pareto archive, so that it only contains non-dominated solutions. Unique non-dominated solutions from current iteration are used as initial solutions for next iteration of parallel neighborhood searches. Thus, the number of starting points per solution in neighborhood search depends from the number of those unique non-dominated solution from previous iteration.

Furthermore, since PPATSs initial solution is chosen at random, it uses a number of initial adaptation runs during which the value of maximum route length time is much bigger than the desired constraint value and is reduced with each iteration until it attains the desired maximum route length per vehicle. During that period non-dominated solutions, unless feasible and meeting the desired constraint, are not added to external set.

5. Computational experiments

In order to compare TS and PPATS we decided to use two types of comparison. The first one takes into account algorithms speed and number of neighborhoods checked, while the second compares results provided by both of those algorithms.

Computational times and number of neighborhoods checked are presented in Tab. 1. Since TS runs on single core, it is limited to 100 neighborhoods checked per run, and for instance sizes of 40 consumer locations and more it is slower than PPATS. For instance, size of 500 nodes, sequential TS is over 14 times slower and explores over 16 times less neighborhoods than PPATS. Moreover, each neighborhood search means checking $\frac{n^2-n}{2}$ solutions, where n is a number of nodes in problems instance.

For each instance size, we also collected the set of Pareto optimal solutions P^A , where $A \in \{TS, PPATS\}$, and determined the set P^* consists of non-dominated solutions of both sets. Finally, for each algorithm A , we

Table 1. Comparison summary of computational time and checked neighborhoods.

Instance size	Single Core Neighborhoods	Single Core Time[s]	Single Core Pareto	Single Core $d(TS)$	N-Cores Neighborhoods	N-Cores Time[s]	N-Cores Pareto	N-Cores $d(PPATS)$	Overall $ P^* $
10	100	0,33	1	1	100	2,12	1	1	1
20	100	0,337	2	1	285	7,767	5	4	5
30	100	1,082	1	0	155	3,919	2	2	2
40	100	10,56	1	0	272	3,925	3	3	3
50	100	12,676	1	0	328	8,719	6	5	5
100	100	39,673	2	1	280	8,703	3	2	2
200	100	263,57	2	0	987	40,949	9	9	9
300	100	942,05	1	0	1162	327,743	8	8	8
400	100	2136,727	1	0	1891	476,716	31	30	30
500	100	4331,783	1	0	1628	298,316	26	26	26

determined the number of solutions $d(A)$ from P^A included in P^* . The number of non-dominated solutions of both algorithms as well as number of elements of every sets are shown in Tab. 1.

Durring our research we used two GPUs, nVidia Tesla S2050 and GeForce® GTX 480, for our PPATS computational experiments. Sequential Tabu Search was tested, for comparison, on Tesla S2050 GPU.

6. Conclusions and further research

As expected, proposed parallel algorithm was able to outperform classic TS algorithm in terms of speed, even though its computational complexity was higher. Moreover, PPATS algorithm dominated almost all solutions found by aforementioned TS algorithm. Applied neighborhood search method was crude at most, and further development of such is in order. Future studies of parallel multi-objective algorithms, apart from Tabu Search and other local search methods, should involve more complex selection methods of initial solutions and types of those searches. Tabu list is in its simplest form, a short-term set of the solutions that have been visited in the recent past. Applying more evolved models will allow us to differentiate neighborhood search methods used.

References

- [1] G. B. Dantzig, J. H. Ramser, The truck dispatching problem, *Management Science* 6 (1) (1959) 80–91.
- [2] M. R. Garey, D. S. Johnson, *A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [3] R. Bowerman, B. Hall, P. Calamai, A multiobjective optimization approach to urban school bus routing : Formulation and solution method, *Transportation Research: Part A, Policy and Practice* 29 (1995) 107–123.
- [4] T.-R. Lee, J.-H. Ueng, A study of vehicle routing problem with load balancing, *International Journal of Physical Distribution and Logistics Management* 29 (10) (1998) 646–657.
- [5] K. Zografos, K. Androustopoulos, A heuristic algorithm for solving hazardous material distribution problems, *European Journal of Operational Research* 152 (2) (2004) 507–519.
- [6] J. Pacheco, R. Martí, Tabu search for a multi-objective routing problem, *Journal of Operational Research Society* 57 (1) (2006) 29–37.
- [7] A. Corberan, E. Fernandez, M. Laguna, R. Martí, Heuristic solutions to the problem of routing school buses with multiple objectives, *Journal of Operational Research Society* 53 (4) (2002) 427–435.
- [8] D. Żelazny, Multicriteria optimization in vehicle routing problem, in: *National Conference of Describe Processes Automation*, 2012, pp. 157–163.
- [9] E. Ulungu, J. Teghem, P. Fortemps, D. Tuytens, Mosa method: A tool for solving moco problems, *Journal of Multi-Criteria Decision Analysis* 8 (4) (1999) 221236.
- [10] T. S. L. Paquete, L. Chiarandini, Pareto local optimum sets in the bi-objective traveling salesman problem: An experimental study, in: *Lecture Notes in Economics and Mathematical Systems*, 2004, pp. 177–200.
- [11] J. Schaffer, Multiple objective optimization with vector evaluated genetic algorithm, in: *Proc. of the Int. Conf. on Genetic Algorithm and their Applications*, 1985, pp. 93–100.
- [12] C. Keller, Multiobjective routing through space and time: The mvp and tdvrp problems, Ph.D. thesis, University of Western Ontario (1985).
- [13] C. Keller, M. Goodchild, The multiobjective vending problem: A generalization of the traveling salesman problem, *Environment and Planning B: Planning and Design* 15 (4) (1988) 447460.
- [14] F. Glover, C. McMillan, The general employee scheduling problem: an integration of ms and ai, *Computers and Operations Research* 13 (5) (1986) 563–573.