

An Evolutionary Approach to Pickup and Delivery Problem with Time Windows

Jean-Charles Créput¹, Abder Koukam¹, Jaroslaw Kozlak^{1,2}, and Jan Lukasiak^{1,2}

¹ University of Technology of Belfort-Montbéliard, Belfort, France

² AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
Jean-Charles.Creput@utbm.fr, kozlak@agh.edu.pl

Abstract. Recently, the quality and the diversity of transport services are more and more required. Moreover, in case of a great deal of services and selling goods, a significant part of price is transport cost. Thus, the design of models and applications which make possible efficient transport planning and scheduling becomes important. A great deal of real transport problems may be modelled by using Pickup and Delivery Problem with Time Windows (PDPTW) and capacity constraints, which is based on the realization of a set of transport requests by a fleet of vehicles with given capacities. Each request is described by pickup and delivery locations, time periods when pickup and delivery operations should be performed and needed load. Application of evolutionary approach has brought good results in case of another, simpler transport problem – the Vehicle Routing Problem with Time Windows (VRPTW). This paper is aimed at proposing a straightforward extension of VRPTW based heuristics for the PDPTW.

1 Introduction

Recently, the importance of passenger and cargo transport services increases. Taking into consideration a growing competition in the domain, companies make an effort to improve the quality of services and to offer new kinds of services, which are able to meet the needs of users.

A lot of transport services may be described by different kinds of vehicle routing problems such as vehicle routing problem (VRP), vehicle routing problem with time windows (VRPTW), pickup and delivery problem (PDP), pickup and delivery problem with time windows (PDPTW). These problems are defined by a set of known transport requests to be performed with the least resources available and time consumption, and with maximum satisfaction of customers. Time windows concern acceptable periods of realization of service in visited points. Maximal capacities of vehicles cannot be exceeded by overall load.

The routes of vehicles should start and finish at a given depot point. In VRPTW it is assumed that each transport request is described by one location which should be visited. The cargoes are being delivered either from starting depot to different destination locations or from different starting locations to one destination location. VRPTW has practical applications like post delivering, newspapers distribution or transport of employee to a workplace.

In PDPTW it is assumed, that each transport request is described by two locations, which are to be visited by the same vehicle: pickup point and delivery point. PDPTW problem has a lot of possible applications [4, 9] such as the transport of elderly and handicapped people, shared taxi services, sealfit and airlift, discharge of larvicide, school bus routing and scheduling.

The problem of finding a solution for VRPTW is better explored than for PDPTW case, there are publications concerning VRPTW subject since the seventies. Here, we focus on PDPTW with capacity constraints. Because evolutionary algorithms (EA) have made possible to obtain good solutions for similar problems like VRPTW, we try to extend their embedded specific heuristic designed for VRPTW to PDPTW in a straightforward way.

The organization of this paper is as follows: section 2 contains a state of the arts of recent attempts to solve PDPTW using modern meta-heuristics. Section 3 presents our evolutionary algorithm. Section 4 describes the experimental results obtained, whereas section 5 concludes.

2 Research Overview

The problem encountered while trying to solve PDPTW is the high computational complexity which causes the calculation time for non-trivial size problems to be very high.

A state of the arts of exact methods as well as heuristic solutions of PDPTW is presented in [3, 4, 10]. Most recent attempts to solve PDPTW problem use meta-heuristics based on tabu search. These approaches consist of two phases: in the first one some initial solutions are generated using heuristics of routes constructions and in the other, these solutions are improved through searching the solution space by introducing changes according to given rules [5, 6, 7, 8, 11].

From analysis of the related works we retain the following two classes:

Heuristic used for generation of initial solutions: The different kinds of algorithms based on insertion of requests to routes are applied. They differ by the method used to define the order of insertions. To obtain more balanced routes, the routes may be initialised by the seed-request – the one which is the farthest from depot point or which time-window of pickup starts first. Request points may be sorted by increasing angle $\angle AOS$ where S is location of request point, O the depot location and A a location which is used as a reference. For each route, only those for which pickup points are situated in an adequate bracket of values of $\angle AOS$ are analysed. Insertions place of request in a route may be selected in different ways. It may be the first feasible place found or the place where cost function increase, after insertion, is the lowest one.

Search in solution space: Some algorithms take into consideration the solutions which do not respect time and capacity constraints. It is worth noting that operations applied in the presented papers, despite different names, have similar algorithms and perform three kinds of actions: removing a request from a route and adding it into another route, exchanging pairs of pickup and delivery points among selected routes as well as changing order of visited points within one route.

We have not found solutions of PDPTW using evolutionary approaches, but EA provide solutions for VRPTW problem which are among the best ones [2]. Retaining

the basic principles of solutions for PDPTW and starting from known heuristics for VRPTW we propose a new evolutionary algorithm for PDPTW.

3 System Model and Optimisation Algorithm

In this section, the evolutionary algorithm for solving PDPTW is described. It is composed of a method of representation of the individual, a fitness function expressing the quality of solution as well as a set of basic evolutionary operators. We try to develop the straightforward EA solution with the simples operators.

Each request point contains following information: type of point (pickup or delivery), times of arrival and departure, occupied vehicle capacity. Times of arrival and departure are defined by two values: \min_T and \max_T which define the maximum time for the vehicle to arrive at their request place and the minimum time unless it can not serve the request. If the vehicle arrives before \min_T , it has to wait at the place.

3.1 Individual and Population

Individual in the evolutionary algorithm is a feasible solution of PDPTW, which consists of a list of routes. One route is associated with one vehicle. Each route consists of a sequence of request points (pickup and delivery points) which are visited by the given vehicle and of the following additional attributes: number of served requests, total travel time, total travel distance, total cost.

Population consists of PDPTW solutions which are modified by evolutionary operators to obtain the next generation of candidate solution.

3.2 Fitness Function

To estimate a quality of solution, we use the fitness function presented by the following equation:

$$F(x) = \frac{\alpha_1}{K} + \frac{\alpha_2}{\sum_{k=1}^K D_k} + \frac{\alpha_3}{\sum_{k=1}^K T_k}$$

where: K – number of vehicles, D_k – total travel distance of k -th vehicle, T_k – total travel time of k -th vehicle, $\alpha_1, \alpha_2, \alpha_3$ – weight parameters.

3.3 Construction of Initial Population of Solutions

The number of solutions in the initial population is equal to 50. We have used one single algorithm to create an initial generation of solutions. It is *Simple Random Insertion Algorithm*, defined as follows:

```

Let RS contains all requests to realize;
while RS not empty begin
    empty route  $VR_i$  for vehicle  $v_i$  is created;
    number_of_attempts=number_of_elements(RS);
    attempt=0;
    repeat
        A request  $r_j$  is randomly selected from RS;
        Attempt to insert  $r_j$  into route  $VR_i$ ;
        if insertion succeeded then
            remove  $r_j$  from RS;
    until (attempt = number_of_attempts)
end

```

3.4 Evolutionary Operators

In the following paragraphs, the principles of the *Reproduction*, *Mutation* and *Selection* operators are presented. Cross-over and mutation operators are inspired by those presented in [12], for VRPTW problem solving.

Reproduction. The individuals are selected to reproduce by a tournament method. An existing population is divided into groups and after that from each group the best individual is selected and added to the parent population. Two methods of creation of offspring from two parents are used: in the first one, an exchange of parts of two routes takes place, in the other, complete routes are exchanged. It is assumed, that after the reproduction operation only one offspring is created. These operators proposed in [12] for VRPTW are extended by considering a single request as a binary pickup and delivery pair.

Sequence-Based Crossover (SBX). The principle of this operator is an exchange of fragments of routes between two solutions. In order to do this, two parents are selected from the population using tournament method, then for each of these solutions, one route is selected. Lists consisting of requests served by each of vehicles are created and an index (within the range from 0 to up to size of shortest from among both the lists) is selected by random. The requests which are on positions lower or equal to the index value in first route or on position higher than index value in the second route are included into a new set of requests. On the basis of this set of requests a new route is constructed. If the creation of route is impossible then operation is rejected, otherwise a new solution is created. This new solution consists of the new route and the routes from replaced parent excluding the one which was changed. Then, the feasibility of the entire solution is examined, and, if necessary, a repair operation is performed. The unfeasibility takes place when:

- Not all the requests are served. In this case, absent requests are inserted into positions which gives best values of fitness function. If it is not possible, then operation is withdrawn and the offspring is abandoned.
- Some requests are served two times. If the same requests appear in the second parent and in the offspring, then they are removed from the parent, if they appear two times in the offspring than one of them is removed.

Route-Based Crossover (RBX). There is made a choice of two individuals from population, then for each of these two, one route is selected. Then, the selected routes are exchanged between individuals. As in the case of SBX operator, so as obtain feasible solution where each request occurs only once, the execution of repairing procedure may be necessary.

Mutation. Two different types of mutation operators are defined, extending the operators for VRPTW presented in [12].

One-level exchange (1M). The role of this mutation is trying to decrease the number of needed vehicles. An individual and a route are selected by random and then it is tried to move all requests served by this route to other routes in such a way, that the obtained solutions are feasible and total riding time of other routes is minimized.

Local search (LSM). The role of operation LSM is to improve the quality of a route by changing the order in which request points are visited.

One route is selected by random. For each requests from that route we try to find a better location inside that route, if there is such a place, we move a request to that place.

Selection. New generation is obtained after modification of previous one by crossover and mutation operators, adding best solution from previous generation or removing best existing or randomly selected solutions. Total number of individuals in the population cannot exceed the upper limit and be under the lower limit.

4 Results

The experiments were performed for the set of PDPTW benchmarks [1]. The main parameters of evolutionary algorithm were set as follows: size of population = 50, number of iterations = 500, weigh parameters of fitness function $\alpha_1=\alpha_2=\alpha_3=1$.

The Table 1. contains a list of evolutionary operators with probabilities of their applications for the given generation, and a frequency of attempts of execution for one generation.

Table 1. Evolutionary operators

Operator	Probabiliy	Iterations	Details
Fitness	1	1	Vehicle number, Travel Time, Travel Distance are considered
PDP Cross Over	1	2	random selection of RBX or SBX, parents selected by tournament
LS PDP Mutation	0.3	3	Local Search for all individuals
M1 PDP Mutation	0.5	2	M1 for randomly selected individuals
Selection	1	1	add the 5 best individuals or remove randomly selected individuals

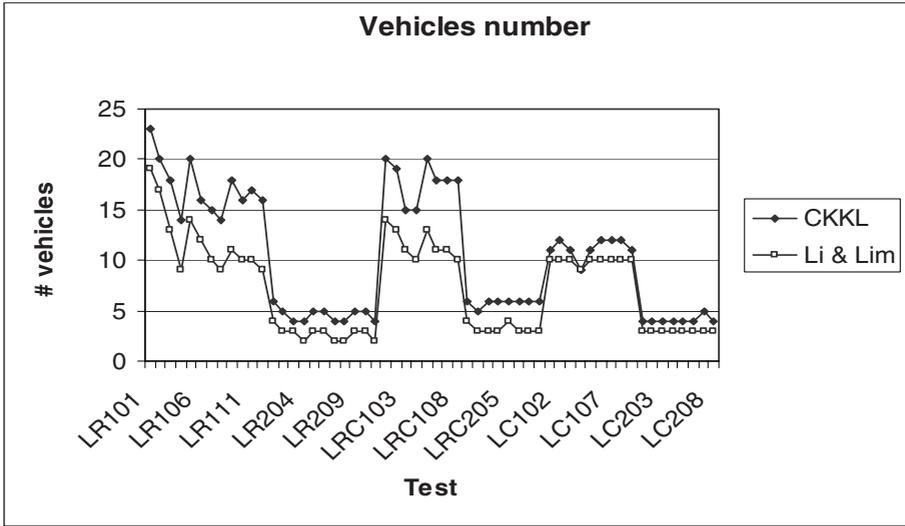


Fig. 1. Number of vehicles

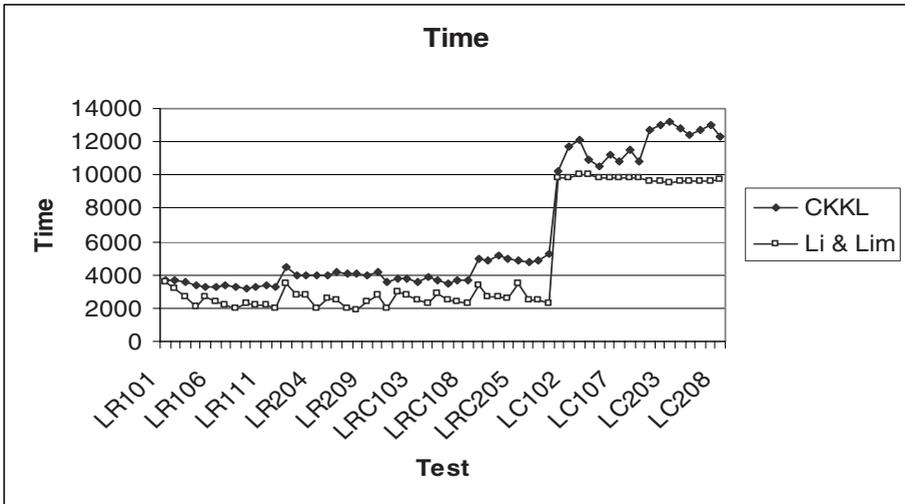


Fig. 2. Total travel times

On Fig. 1 and Fig. 2 are presented vehicle number and total travel time obtained by us [CKKL] and the ones accessible on WWW [1], obtained by Li & Lim [7]. On the X-axis the names of benchmarks are marked.

All requests are satisfied but the obtained results (number of used vehicles, total travel time, total distance) are usually worse about few dozens of percents than the best known ones.

5 Conclusions

In this article, an example of evolutionary approach to solve PDPTW was presented. It is a straightforward extension of VRPTW heuristic approach. Additionally, experiments and modifications including an adjustment of configuration parameters and some modifications of evolutionary operators are necessary. Results are suboptimal in regard to bests [Li & Lim] results, however we can see some concordance of the global shape of the plots with the [Li & Lim] results. It is explained by the simple procedure we have achieved in our approach. Thus, this first prototype would certainly be improved by adjustment of operator parameters and introduction of more problem specific knowledge.

References

1. Benchmarks – Vehicle Routing and Travelling Salesperson Problems. <http://www.sintef.no/static/am/opti/projects/top/vrp/benchmarks.html>, Updated; 23 April 2003.
2. Braysy, O.: Genetic Algorithms for the Vehicle Routing Problem with Time Windows. *Arpakauus 1*, special issue on Bioinformatics and Genetic Algorithms, 2001.
3. Cordeau, J.-F., Laporte, G.: The Dial-a-Ride Problem: Variants, Modeling Issues and Algorithms. *Les Cahiers du GERAD*, 2002.
4. Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., Soumis, F.: The VRP with Pickup and Delivery. *Les Cahiers du GERAD*, 2000.
5. Gendreau, A., Guertin, F., Potvin, J.Y., and Séguin, R.: Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Rapport technique CRT-98-10*, Université de Montréal, 1998.
6. Lau, H.C., Liang, Z.: Pickup and Delivery with Time Windows : Algorithms and Test Case Generation. *Proceedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*, Dallas, USA, 2001.
7. Li H., Lim, A.: A Metaheuristic for the Pickup and Delivery Problem with Time Windows. *Proceedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*, Dallas, USA, 2001.
8. Lim, H., Lim, A., Rodrigues, B.: Solving the Pick up and Delivery Problem using “Squeaky Wheel” Optimization with Local Search. *Proceedings of American Conference on Information Systems, AMCIS 2002*, Dallas, USA, 2002.
9. Madsen, O.B.G., Ravn, H.F., Rygaard, J.M.: A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60, 193-208, 1995.
10. Mitrowic-Minic, S.: Pickup and Delivery Problem with Time Windows: A Survey. SFU CMPT TR. 1998-12, <ftp://fas.sfu.ca/pub/cs/techreports/1998>, 1998.
11. Nanry, W.P., Barnes, J.W.: Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B* 34, Elsevier Science Ltd, pages 107-121, 2000.
12. Potvin, J. Y., Bengio, S.: The vehicle routing problem with time windows - Part II: Genetic search. *INFORMS Journal on Computing* 8, pp. 165-172, 1996.