An Evolutionary Algorithm for Vehicle Routing Problem with Real Life Constraints

Giuseppe Confessore[1], Graziano Galiano[1] and Giuseppe Stecca[1]
[1] Institute of Industrial Technologies and Automation, National Research Council, Italy

Abstract
Real life distribution problems present high degree of complexity mostly derived by the need to respect a variety of constraints. Moreover they are not considered by the classical models of the vehicle routing literature. In this paper we consider a vehicle routing problem with heterogeneous vehicle fleet with different capacity, multi-dimensional capacity constraints, order/vehicle, item/vehicle, and item/item compatibility constraints, different start and end locations for vehicles, and multiple time windows restrictions. We propose an evolutionary algorithm based on the combination of a genetic algorithm and local search heuristics. We investigated the performance of the implemented algorithm in the large-scale retail and in the waste collection industries.

Keywords:
Vehicle Routing Problem; Distribution Process; Evolutionary algorithm

1 INTRODUCTION

Real-life distribution processes are affected by high complexity due to elevated number of constraints to respect, to different optimisation criteria, to the need of practical extensions, and to need for responsiveness.

This work treats a dynamic real-life vehicle routing problem proposing an algorithm to solve a distribution problem in a manufacturing system. A basic distribution problem consists of a set of customers requiring the delivery of goods within given time windows and deliveries are done with capacitated vehicles departing from and returning to a depot. The goal consists in minimizing the number of necessary vehicles to effectuate the routing, or the total transportation cost (e.g. total distance covered by the set of vehicles). This is a vehicle routing problem that, although well known in the literature and very studied in its different variants, is often subject of strong simplifications not much suitable to the real world where the problem involves several real constraints.

In real-life problems we have a number of complexities that are not considered by the classical models found in the literature. In this paper we consider a generalised vehicle routing problem (CVRPTW) with a diversity of practical constraints. Among those are multi time window restrictions, a heterogeneous vehicle fleet with different capacity, travel times, temporal availability, travel costs, order/vehicle compatibility constraints, customers with multiple orders of pickup and delivery, different start and end locations for vehicles, route restrictions associated to orders, customers and vehicles, and drivers' working hours.

We propose an evolutionary approach based on the hybridisation of a genetic algorithm with insertion heuristic techniques. This algorithm was implemented and included as optimization module in a software product used by some companies in the transportation planning. We have had the opportunity to really validate the behavior of the module and to provide the results of the improvements achieved in three different industrial scenarios in which the software was installed: large-scale retail industry, waste collection and maritime transportation. In each scenario our algorithm has proven to be very efficient.

2 THE REAL LIFE PROBLEM

2.1 CVRPTW

The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) is a well known strongly *NP*-hard problem, and it is a generalization of the Capacitated Vehicle Routing Problem (CVRP). The CVRP consists of finding a collection of simple tours of minimum cost in a connected digraph starting from and ending to a common depot, such that each customer (i.e., a node of the digraph) is visited exactly by a tour, and the sum of the demands of the customers visited by a tour does not exceed the vehicle capacity. In general the objective is to find the minimum number k of tours; a secondary objective is often either to minimize the total travelled distance or the duration of the tours. In the CVRPTW, service at each customer must start within an associated time window and the vehicle must remain at the customer location during service. If a vehicle arrives before the customer is ready to begin the service, it waits.

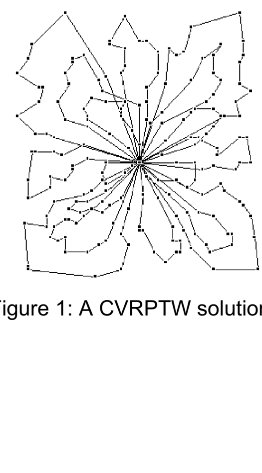

Figure 1: A CVRPTW solution.

The CVRPTW is one of the most studied variations of the VRP and recent surveys can be found in [1-5].

2.2 Practical constraints

A transportation order is specified by a customer location which has to be visited in a particular sequence by the same vehicle. The vehicles can have different capacities, as well as different travel times and travel costs between locations, and other compatibility characteristics which can constrain the possibility of assigning the transportation requests to certain vehicles. Instead of assuming that each vehicle becomes available at a central depot, each vehicle is given a start location where it becomes available at a specific time. All locations have to be visited within time windows different by type of transportation order.

A tour of a vehicle is a journey starting at the vehicles start location and ending at its final location, passing all other locations the vehicle has to visit in the correct sequence. For each tour we have additional constraints:

- Maximum number of deliveries and picks;
- Maximum time between subsequent of deliveries or picks;
- Maximum wait time before begin of service;
- Vehicle reuse for more tours but subject to restrictions to drivers' working hours.

Real-life problems often require rich models, in most of the literature on routing problems however, some simplifying assumptions are made. A discussion of real life vehicle routing can be found in [6-8].

2.3 Feasibility and unfeasibility

A tour is feasible if and only if for all request assigned to the tour compatibility constraints hold and at each location in the tour time window and capacity and additional restrictions hold. In our approach we consider both feasible and unfeasible solutions.

An unfeasible solution is penalized in proportion to the size of its constraint violations. The purpose of a penalty function formulation is to produce a representation of the problem that can be directly and naturally encoded as a genetic algorithm. Let *x* be a solution to a minimization constrained optimization problem. The objective of the problem is defined as:

$$\min_{x_i} z = Z(x) + P(x) \tag{1}$$

where $Z(x)$ is the objective function value produced by *x* and $P(x)$ is some total penalty associated with constraint violations at x.

Typically, the penalty imposed on an unfeasible solution will severely reduce the fitness of the solution in question, leading to quick elimination of the solution from the population. This may be undesirable, since unfeasible solutions may carry valuable information and may be useful in searching for optimal values. We use an evolutionary process evolving two subpopulations of solutions: the feasible subpopulation (consisting only of feasible solutions) and the unfeasible subpopulation (consisting only of unfeasible solutions). In the process, feasible solutions may produce unfeasible ones and unfeasible solutions may produce feasible ones.

It is evident in [9-11] that this technique has considerable merits.

At the end of the search process a set of no dominated solutions are found. The solving method is able to find a good solution with no violation and a set of efficient (no dominated) solutions in terms of travelling cost and customer service level.

3 AN EVOLUTIONARY APPROACH

3.1 Evolutionary algorithms

Genetic algorithm (GA) is an adaptive heuristic search method based on population genetics [12,13]. GA evolves a population of individuals encoded as chromosomes by creating new generations of offspring through an iterative process until some convergence criteria or conditions are met. The best chromosome generated is then decoded, providing the corresponding solution. At each iteration, the creation of a new generation of individuals involves primarily three major steps or phases: selection, recombination and mutation. The selection phase consists in choosing randomly two (parent) individuals from the population with a probability, in general, proportional to the fitness (goodness) of the individuals in order to emphasize genetic quality while maintaining genetic diversity. The recombination (i.e., reproduction or crossover) process makes use of genes of selected parents to produce offspring that will form part of the next generation. The mutation consists in randomly modifying gene(s) of a single chromosome (individual) at a time, to further explore the solution space and ensure or preserve genetic diversity. Both recombination and mutation operators are randomly applied with given probabilities.

Hybrid genetic algorithms combine the above scheme with heuristic methods for further improving solution quality.

There are also many applications of evolutionary techniques to the VRP and its variants. However, when applied alone, their success is limited. This led researchers to rely on hybrid approaches that combine the power of an evolutionary algorithm with the use of specific heuristics or to simplify the problem.

3.2 Description of our approach

Our hybrid genetic algorithm works on a population composed by a subpopulation of feasible solutions and a subpopulation of unfeasible solutions.

The schema of algorithm is as follow:

Definition: population *P* formed by *n* solutions and composed of $r \cdot n$ feasible solutions and $(1-r) \cdot n$ unfeasible solutions.

Initial Population: Fill the set *P* with solutions obtained by the randomized version of I1 Heuristics.

Selection: select solutions from *P* using a biased roulette wheel; a control mechanism is applied in order to maintain a prefixed rate *r* of feasible individuals on population *P*.

Crossover: The hybrid sequence based crossover (HSBX) is applied on two selected solutions.

Mutation Phase: Apply mutation operator *i* with a probability pm_i on a solution.

Return: the best solution.

The algorithm, starting from an initial population, progressively evolves the solutions by recombining feasible and unfeasible ones.

The considered selection operator is a fitness-proportional selector. Crossover and mutation operators are hybridized with insertion heuristics. In particular we use an extension of the SBX crossover operator.

Individuals are initially generated by a randomized version of the heuristic I1 proposed in [14]. The randomization is considered both in the seed computation to initialize a new tour and in the best feasible insertion position. Elitism strategy is implemented.

3.3 Crossover and mutation operators

We implemented a hybridisation of the Sequence-Based Crossover *SBX* described in [15] in which tours of parent individuals are merged. Given a pair (X, Y) of individuals, the crossover operator $HSBX(X, Y)$ applied on the pair (X, Y) of solutions produces an offspring X'. We have also considered four different hybridised mutation operators. The description of these operators is in our previous work [16].

3.4 Fitness function

A solution x is specified by a pair $<z, p>$ where z is the objective function value and p is the total penalty associated with constraint violations at x. Let x_1 and x_2 be two solutions, we have:

1. If $z_1 < z_2$ and $p_1 \leq p_2$ then x_1 is better than x_2
2. If $z_1 > z_2$ and $p_1 \geq p_2$ then x_2 is better than x_1
3. If $z_1 < z_2$ and $p_1 > p_2$ then "which solution is the best?"
4. If $z_1 > z_2$ and $p_1 < p_2$ then "which solution is the best?"

For the case 3 and 4 we introduce the ranges d_1 and d_2, and the thresholds s_1 and s_2 considering the following rules to determine the best solution:

- If $|z_1 - z_2| \leq d_1$: the solution with the lowest p value is the best;
- If $|z_1 - z_2| > d_1$ and $\max(p_1, p_2) \leq s_1$: the solution with the lowest z value is the best;
- If $d_1 < |z_1 - z_2| \leq d_2$ and $s_1 < \max(p_1, p_2) \leq s_2$: the solution with the lowest p value is the best;
- If $|z_1 - z_2| > d_1$ and $\max(p_1, p_2) \leq s_1$: the solution with the lowest z value is the best;
- If $\max(p_1, p_2) > s_2$: the solution with the lowest p value is the best.

For example in Figure 2 we have: the solution x_2 is better than x_1 , x_1 is better than x_3 and x_2 is better than x_3.

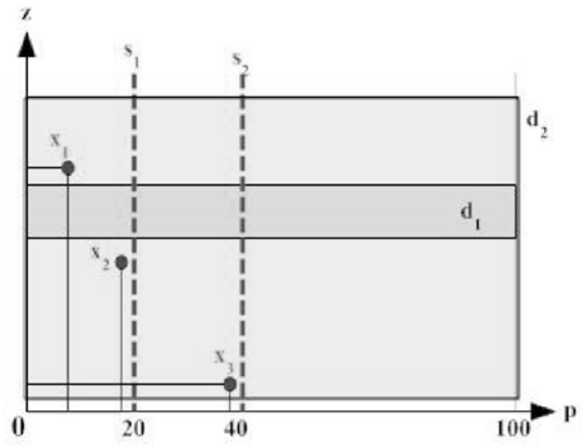

Figure 2: Example of selection of the best solution.

4 CONCLUSIONS

In this paper a vehicle routing problem with heterogeneous vehicle fleet with different capacity, multi-dimensional capacity constraints, order/vehicle, item/vehicle, and item/item compatibility constraints, different start and end locations for vehicles, and multiple time windows restrictions has been presented. The solving algorithm is a hybrid genetic algorithm with different hybrid crossover and mutation operators.

The hybrid genetic algorithm has been implemented in a software tool that allow easy configuration of objectives and constraints. It has been found that the proposed tool is effective and useful for more variants of Vehicle Routing Problem.

Computational experiments are performed on test cases derived from the real-life problem. They have shown that the algorithms perform well for problems with hundreds of vehicles and several hundreds of transportation requests. The combination of fast response times and the capability of handling the practical complexities allow the use of our algorithms in dynamic routing systems.

The solutions obtained by the proposed approach for various versions of the problem, in order to achieve effective use in real environments, are going to be presented in the future works.

5 REFERENCES

[1] Toth P. and D. Vigo (2000), *The Vehicle Routing Problem*, SIAM monographs on Discrete Mathematics and Applications.

[2] Bräysy O. and M. Gendreau (2003a), "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", Forthcoming in *Transportation Science*.

[3] Bräysy O. and M. Gendreau (2003b), "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics", Forthcoming in *Transportation Science*.

[4] Laporte G., M. Gendreau, J.Y. Potvin and F. Semet (2000), "Classical and Modern heuristics for the vehicle routing problem", *International Transactions In Operational Research* 7, 285-300.

[5] Tan, K.C., L.H. Lee and K. Q. Zhu (2000), "Heuristic methods for vehicle routing problem with time windows", *Artificial Intelligence in Engineering* 15, 281-295.

[6] Psaraftis HN (1988). Dynamic vehicle routing problems. In B.L. Golden and A.A. Assad, editors, Vehicle routing: Methods and studies, pages 233-248. North-Holland Amsterdam.

[7] Psaraftis HN (1995). Dynamic vehicle routing: Status and prospects. Annals of Operations Research, 61:143-164.

[8] Savelsbergh MWP and Sol M (1998). DRIVE: dynamic routing of independent vehicles. Operations Research, 46:474-490.

[9] Kimbrough, S.O., Lu, M., Wood, D.H., Wu, D.J.: Exploring a two-market genetic algorithm. In Langdon, W.B., Cantù-Paz, E., et al., eds.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), San Francisco, CA, Morgan Kaufmann (2002) 415–21

[10] Kimbrough, S.O., Lu, M., Wood, D.H., Wu, D.J.: Exploring a two-population genetic algorithm. In Cantù-Paz, E., et al., eds.: Genetic and Evolutionary Computation (GECCO 2003). LNCS 2723, Berlin, Germany, Springer (2003) 1148–1159

[11] Kimbrough, S.O., Lu, M., Safavi, S.M.: Exploring a financial product model with atwo-population genetic algorithm. In: Proceedings of the 2004 Congress on Evolutionary Computation, Piscataway, NJ, IEEE Neural Network Society, IEEE Service Center (June 19–23, 2004) 855–862

[12] Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA.

[13] Holland, J.H. (1975), “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor.

[14] Solomon, M.M. (1987), “Algorithm for the Vehicle Routing and Scheduling Problems with Time Window Constraints”, *Operation Research* 35, 254-265.

[15] Potvin, J.-Y. and S. Bengio (1996), “The Vehicle Routing Problem with Time Windows Part II: Genetic Search”, *INFORMS Journal on Computing* 8, 165−172.

[16] Confessore, G., Galiano, G., Stecca, G., 2006, General Purpose Application For Vehicle Scheduling And Routing In Complex Distribution Environments, 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, eds. Roberto Teti, pp 299-302, ISBN 88-95028-01-5.