AN HYBRID EVOLUTIONARY ALGORITHM FOR STOCHASTIC VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS

ROBINSON A. JAQUE P.

ID: 299800

Thesis presented as a partial requirement for the degree of

MASTER OF SCIENCES

COMPUTER SCIENCE

Advisor:
GERMÁN HERNÁNDEZ, PH.D.
Associated Professor

NATIONAL UNIVERSITY OF COLOMBIA
SCHOOL OF ENGINEERING
DEPARTAMENT OF COMPUTER AND INDUSTRIAL ENGINEERING
BOGOTA D.C.
2011

Approved by the School of Engineering in fulfillment of the requirements to grant the title of **Master of Sciences — Computer Science**

Germán Hernández, Ph.D.
Advisor of the Thesis

1, Ph.D.
Member of the Jury

2, Ph.D.
Member of the Jury

National University of Colombia
Bogota D.C., Junio of 2011

ABSTRACT

AN HYBRID EVOLUTIONARY ALGORITHM FOR STOCHASTIC VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS

by

ROBINSON A. JAQUE P.
Master of Sciences in Computer Science

NATIONAL UNIVERSITY OF COLOMBIA

Advisor: Germán Hernández, Ph.D.

In this work we propose a hybrid dynamic programming-evolutionary algorithm to solve the vehicle routing problem with stochastic demands, it is a well known NP-hard problem where uncertainty enhances the computational efforts required to obtain a feasible and near-optimal solution. We develop an evolutionary technique where a rollout dynamic programming algorithm is applied as local search method to improve the quality of solutions. Motivated by computational considerations, the rollout algorithm can be applied partially, so, this finds competitive solutions in large instances for which the global rollout dynamic programming strategy is time unfeasible.

ACKNOWLEDGMENTS

Last thing to do :-)

DEDICATION

To Hari

Contents

List of Figures

List of Tables

Notation

Symbol	Definition
n	Number of customers
d_{ij}	Distance between a pair of customers i and j
$E[L_\tau]$	Expected distance of an apriori solution (base sequence) τ
l	Location of vehicle
q_l	Vehicle capacity when customer l has already been served.
$p_i(j)$	Probability of customer i take the demand value j
K	Maximal demand
$\bar{D}_i$	Maximal demand of customer i
Q	Maximal capacity of vehicle

CHAPTER 1

Introduction

1.1. Introduction

The classic problem about vehicle routing (VRP) is an well known *NP-hard* [**?**] optimization problem of importance in different logistics, it consist in delivery goods to a set of costumers geographically dispersed, using a fleet of vehicles that begin its route in the central depot. The problem consists in to assign to each vehicle a route with the objective of minimizing the transportation cost.

The cost generated in the transport of goods, like the size of the fleet of vehicles maintenance, combustible, and so on, are significant owing to the transport processes are involved at all stages of production processes, accounting for 10% to 20% of the final product cost [**33**].

One of the first investigation that studied the vehicle routing problem was in the year 1959, in that work Dantzing and Ramser [**11**] analyzed an oil dispatching problem with trucks, that problem arise as a generalization of the classic traveling salesman problem (TSP), where the salesman has to visit a set of costumers only one time, and then come back to the origin point, building a Hamiltonian road on the graph consisting of customers (vertices) and the possible paths between clients (edges).

Different variations of the VRP have been proposed with the aim of approaching the problem real contexts; these problems include the addition of variables and constraints. The figure below shows a diagram with the most popular variants of VRP.

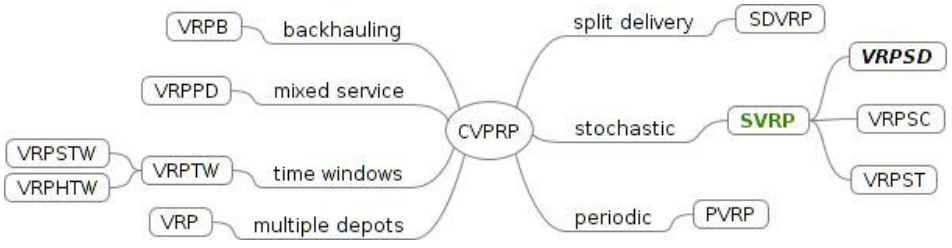

FIGURE 1.1.1. Basic variants of the Vehicle Routing Problem

When vehicles capacity is fixed origin the CVRP (Capacited Vehicle Routing Problem); if there are many depots, then we have MDVRP (Multiple Deposit Vehicle Routing

Problem). The SDVRP (Split delivery Vehicle routing problem) is a relaxation of the problem that allows a costumer be served by several vehicles, being important in cases where the customer demand exceeds the vehicle capacity. Generally the VRP provide a planning for a fixed period, the PVRP (Periodic Vehicle Routing Problem) provides that planning for m periods.

One of the most studied variants of the problem is caused by including time windows for deliveries, VRPTW (Vehicle Routing Problem with Time Windows), for this problem can be considered hard time windows (VRTHTW) that can't be done delivery outside the established periods, and soft time windows (VRPSTW) in which deliveries can be made outside these periods with a penalty.

The problems that include pick-up and delivery can be divided in VRPB (Vehicle Routing Problem with Backhauls) and VRPPD (Vehicle Routing Problem with Pick-Up and Delivery) in the VRPB the group of costumers is divided in two subgroups, for the first group all pick-up's are made of some product, then the vehicle return to the depot, then the deliveries are realized to the second group of costumers. In the VRPPD the pick-up's and deliveries to customers are made simultaneously.

The SVRP (Stochastic vehicle routing problem) arise when there is uncertainty about some of the components of the VRP, i.e. one or more variables are random, usually these problems include VRPSC (Vehicle routing problem with stochastic costumers) that include random customers, the VRPST (Vehicle Routing Problem with Stochastic Times) where travel times are random and VRPSD (Vehicle Routing Problem with Stochastic Demands), where the customers' demands are known only with a probability distribution.

The SVRP's differ from the deterministic VRP in many important aspects. The solution concept is different, many properties of the deterministic problem are not manageable in the stochastic case and solution methodologies are considerably more complex, often SVRP is considered a computationally intractable problem and only small instances can be solved optimally and algorithms are difficult to design and evaluate **[19]**.

The SVRP is either often modeled in the framework of stochastic programming (optimization) integer or mixed or as a Markov decision process. In stochastic programming, problems are usually modeled as two-stage usually as chance constrained program (CCP) or as a stochastic program with recourse (SPR).

The VRPSD is an open problem of great importance in logistics owing to the diversity of real situations it represents, this problem occur in the delivery of home heating oil **[13]** in which each customer maintains a local inventory of the product and consumes

a amount of oil each day, therefore each day a fleet of trucks is dispatched to resupply a subset of customers, stochastic demands are also evident in the collection of money by the vehicles of values, e.g. is touched off when collect money by a central bank [**16**] from several but not all of its branches every day, the capacity of the vehicle used may be constrained for an upper bound on the amount of money that a vehicle might carry for safety reasons. The distribution of demand at each certain branch may be different, associated with the amount of money it handles. Other VRPSD arise in delivering the post to large customers [**21**], vending machines [**34**] or delivering medical supplies in response to large-scale emergency [**12**], in recycling and waste management, among others.

1.2. Proposal

Design a hybrid evolutionary algorithm which combine stochastic dynamic optimization operators to find solutions for the vehicle routing problem with stochastic demands, moreover evaluate the algorithm performance.

1.3. Objective

Design and test a hybrid evolutionary algorithm + stochastic dynamic optimization (SDO) operators for the vehicle routing problem with stochastic demands.

1.3.1. Specifics Objectives

(1) Analyze the alternatives for modeling and representing problem, and selected the most computationally convenient.
(2) Design evolutionary and stochastic dynamic optimization operators algorithm to solve VRPSD.
(3) Implement the designed algorithm.
(4) Select benchmarking problems instances and alternative algorithms for testing.
(5) Develop experimental analyses and comparisons.

1.4. Contributions

The evolutionary algorithms haven't been broadly used to solve the vehicle routing problem with stochastic demands, thus a hybrid evolutionary algorithm which combine stochastic dynamic optimization operators is proposed, it contribute with a new methodology to deal with the problem.

1.5. Outline

In this first chapter was carried out a summary of the issues to be addressed, in chapter 2 presents the background and is presented the state of the art of the VRPSD, chapter 3 presents the Methodology used and presents the proposed algorithm, the chapter 4 presents the experiments, the numerical results and compare and discuss the results comparing with the outcomes obtained by other authors, finally in chapter 5 presents the conclusions of the study.

CHAPTER 2

Background

In this chapter we carry out a literature review of VRPSD, applications and methodologies used to deal with it are point out. Moreover, we examine modeling approaches to this problem in order to exploit the structure and solution properties following the methodology proposed, focusing on the stochastic dynamic programming (SDP) approach where first we show a dynamic programming background before of present the SDP formulation for VRPSD.

2.1. A Review of Vehicle Routing Problem with Stochastic Demands

In recent years the literature related to the stochastic vehicle routing problem has been growth owing to its application to the real life problems, as well as the academic interest in treat with the theoretical problem.Consequently, a number of solutions approach have been proposed to deal with these problems.

2.1.1. Application cases

There are many applications of the VRPSD, in a huge number of real situations the customer demand is unknown and only can be estimated its probability distribution. Even, in spite of that the demand can be known in many cases, when the vehicle arrives to the customer the demand value changes. e.g. Delivering petroleum products or industrial gases (Chepuri and Homem-De-Mello [**8**]). To ilustrate these cases we regard the problem where gas stations are placed geographically dispersed and a fuel transport vehicle is entrusted to deliver a fuel quantity determined when the vehicle arrives at the customer location, the demand is unknown because spite of that the remaining fuel was known at the time to make the order, when the vehicle arrive a time after the fuel stock is lesser.

In the case of ATM (automatic teller machines) not only the daily demand for cash is uncertain, but also the maximal amount of cash that may be picked up by trucks for money transportation is limited for security and insurance reasons.

The traveling repairman Problem (TRP)

That problem was introduced by Bertsimas and Van Ryzin [**6**] in 1991, they analyzed the mathematical model for dynamic and stochastic VRP with the dynamic TRP. In that model the objective is to minimize the average time demands in the system.

That problem is presented when a machine breaks down and must be repaired, for to get that objective is important to know the distance between the starting point and the arrival point, the urgency of the situation (it could be hight or low), the availability of the repairman etc. A real example of that problem is the electric power company, that has to travel point by point to fix the problems that can arise suddenly, that problem has the characteristics of increased costumer, and the time fixing the problems variate according to the particular environment.

Many authors have studied the problem, Jothi and Raghavachari [**22**] analyze two algorithms to solve the problem, in 1993 Das and Wortman [**31**] studied the TRP with a single repairman, they proposed a probability model that is useful for evaluating the system performance measures, like the inactivity time, the availability of a machine and of the repairman.

Currier mail services

Now there are many companies that offer the service of pick-up packages or mail and delivery that goods at the point where the costumer indicates, that problems returns dynamic because the driver doesn't know where will be the next pick-up or quantity or volume of goods.

In 2004 Timon et al [**32**], investigated the DVRP (Dynamic VRP) applied to electronic commerce (EC), they say that the environment of the EC has voluminous, unpredictable, and dynamically changing customer orders, They studied the problem Business to Costumer (B2C), in an EC environment, and the purpose was a solution in three phases: initial-routes formation, inter-routes improvement, and intra-route improvement.

Alan Slater [**30**] purpose and routing and scheduling method for the e-commerce environment, and allows to the costumers select their own delivery time windows. The methodology that he used is based in parallel tour-building and parallel insertion algorithms, the confirmation to the costumers is realized using GPS tracking and tracing.

Emergency services

Some services that arise into our society, are "emergency services", when a person is sick and needs and ambulance, when a person is robed and needs the police, in the case of a fire and a person has to call the firefighters, these are examples of services that can arise suddenly in any moment, in that problem is important to minimize the distance between the origin and arrival point, the availability of the resources (cars, tracks etc.), the level of the urgency.

Taxi cab services

The taxi service is another application of the DVRP, is difficult to planning the taxi route before the taxi get out of the taxi's central, because suddenly the costumers need the service , that problem of assign a taxi route is more complex depending of the number of costumers, the positions of the taxi, the time and the arrival point.

Other VRPSD arise in delivering the post to large customers [**21**], vending machines [**34**] or delivering medical supplies in response to large-scale emergency (Dessouky et al., 2005 [**12**]), also is possible view the features of the problem in recycling and waste management, among others.

2.1.2. Solution methods

There are two different ways for dealing with VRPSD, either with fixed routes or a dynamic approach called often reoptimization. The methodology used to solve it depends of the type of model used for represent the problem, in the section 2.2 we point out some mainly formulations useds to address the problem.

Some solution methods find the optimal solution, however the use of these techniques is limited to small sizes problems. Therefore, others methods have been proposed to aproximate optimal solutions.

2.1.2.1. *Exact methods*

The relevant exacts methods are *branch-and-bound* algorithms, the problem can be formulated as a lineal model, Laporte et. al [**23**] propose a *L-shape method* for solve VRPSD, it is an *branch-and-cut* algorithm.

Bernard et al [**9**], present a Dantzing and Wolf decomposition to resolve a dynamic model analyzing multi-periodic vehicles fleet size and modeling the fleet size, of each one. At the end the authors obtain optimal solutions for different demand distribution.

In 2007 Christian H et al [**10**], presented a Branch and price algorithm for the capacitated vehicle routing problem with stochastic demands, they used dynamic programming for to solve a subproblem.

2.1.2.2. *Aproximate methods*

The cyclic heuristic was proposed by Bertsimas [**5**], it is a simple and inexpensive algorithm that guarantee to reach the final state. We explain it in the section 3.6.1.1 where is used in our methodology. Gans and Ryzin [**17**] have studied the dynamic vehicle dispatching systems where the congestion is the main measure of performance, they used a lower and upper bound, basing in a simple batching heuristic, they found stability conditions for the optimal work in heavy traffic.

Yang et. al. [**34**] test two heuristic algorithms which solve the problem in two stages, the *route-first-cluster-next* and *cluster-first-route-next*, they cluster the customers first and finding the best route for each cluster after. Furthermore, Cross-entropy heuristic method is propossed by Chepuri and Homem-de-Mello [**8**] and to estimate the expected distance they use monte carlo sampling.

Local search heuristic commonly used for TSP have been used to solve VRPSD as OrOpt ([**34**], [**7**]), Yang et. al. used the 3-opt local search algorithm too.

Moretti et. al. [**25**] (2009) implemented an algorithm to solve the DVRP, they consider the demand and the location a stochastic variables, and the time windows are soft, their objective is define a set of routes that are dynamically updated, taking into account the new costumers, to solve they used a constructive algorithm with an adaptive tabu search framework.

A genetic algorithm is another option used to solve this kind of problems, Haghani and Jung [**20**] presented a formulation for the DVRP with pick up and delivery and soft time windows where there is multiple vehicles with different capacities, the real time service and and travel times are variant they proposed a genetic algorithm and their results were compared with exact methods.

2.1.2.3. *Dynamic programming*

The dynamic programming is one of the techniques most used to solve this kind of problems, Bertsekas [**2**] proposed efficient methods like Markov decision analysis, linear control model with quadratic cost, policies in the stochastic inventory problem, and so on. Secomandi [**29**] compare neuro-dynamic programming algorithms for VRPSD.

Bertsekas [**4**] present the *rollout algorithm* (RA) applied to combinatorial optimization problems, later Secomandi ([**28**], [**27**]) show up it applied to VRPSD.

Novoa and Storer [**26**] proposed a solution for the VRPSD using dynamic programing algorithms, they also consider the cost-of-go with the help of Monte Carlo simulation, the showed that in that king of problems the best method found is the one-step roll-out that started with a stochastic base sequence.

2.1.2.4. *Hybrid methods*

Bianchi et. al. [**7**] implement a simulated annealing, tabu search, iterated local search, ant colony optimization and evolutionary algorithms, and combine these with 2 local search tecniques OrOpt and 3-opt, the second local search has been used with good results in TSP.

Mendoza et. al. [**24**] proposse a memetic algorithm for the multi-comparment vehicle routing problem with stochastic demands (MC-VRPSD), it's modeled as stochastic programming with resource and under each iteration of the genetic algorithm a 2-opt local search is performed, the results are compared with the deterministic version of the problem.

2.2. Formulation of VRPSD

Defined a graph $G(V, E)$, as $V = \{0, 1, 2, \ldots, n\}$, where the node 0 denote the starting point for vehicles (depot), and the remaining nodes represent individual customers, the set E of edges in the graph represents the roads or paths between a pair of customers (i, j) and d_{ij} is the distance between them, which is assumed to be known, symmetric and satisfies the triangle inequality.

A vehicle (only one) with fixed capacity $Q < \infty$ starts from the depot and executes deliveries (or pickups only) of a product to different customers, D_i denotes the random variable representing the demand of customer i, the probability distribution D_i is discrete and known and is denoted by $p_i(k) = Pr\{D_i = k\}, k = 0, 1, \ldots, K \leq R$. The customer demands are assumed to be independent and its exact value is known only when the vehicle arrives to the customer location, if customer demand exceeds the available capacity of the vehicle, i.e. a *route failure*, the vehicle must return to the depot to restore its original capacity. Hence, the depot must have a capacity at least equal to nR.

Yang, et. al. [**34**] proposse a simple resource action for early replenishment where the vehicle come back to the depot even whenever it haven't depleted its stock to restore the capacity to Q, allowing proactive depot trips to avoid route failures. Hence, considering proactive restocking of the vehicle is not necesary to consider multiple routes, in fact, Yang, et. al. [**34**] point out that a single route is more efficient than multiple vehicle

route system, assuming that only distance constrain the route, ommitting for example time duration.

The objective is minimize the expected distance finding out a routing solution, maybe in the form of routing rules, such that demand at each customer is satisïňĄed. Thus, VRPSDs are usually modeled as mixed or pure integer stochastic programs, or as Markov decision process.

2.2.1. Stochastic programming

The stochastic programming goal is to find an optimal decision in problems that involve uncertainty in the data. Stochastic VRPs can be cast within the frame-work of stochastic programming [**19**]. Stochastic programs are modeled in two stages. In a first stage, *a priori* solution is determined, the realizations of the random variables are then disclosed and, in a second stage, a recourse or corrective action is then applied to the first stage solution. The recourse usually generates a cost or a saving that may have to be considered when designing the first stage solution. A stochastic program is usually modeled either as a chance constrained program (CCP) or as a stochastic program with recourse (SPR).

2.2.1.1. *Chance-constrained programming*

In CCPs, one seeks a first stage solution for which the probability of failure is constrained to be below a certain threshold. A CCP solution does not take into account the cost of corrective actions in case of failure. Mainly, for a given customer demands parameters, i.e. demands parameters (means, variances), one subjectively especifies a control probability looking for avoid that a *route fail*, following [**15**] a VRPSD is formulated as

$$\text{minimize } \sum_{v}\sum_{i,j} d_{ij}x_{ij}^{v} \tag{2.2.1}$$

$$\text{subject to } Pr\{\sum_{i,j} D_i x_{ij}^{v} \leq Q\} \geq 1-\alpha, \forall v = 1,\ldots,NV, \tag{2.2.2}$$

$$x = [x_{ij}^{v}] \in S_{NV} \tag{2.2.3}$$

where $x_i j^v$ is a binary decision variable that takes the value 1 if vehicle v travels directly from customer i to j and 0 otherwise, S_{NV} is the set of feasible routes for the traveling salesman problem (TSP) with NV salesmen.

These models are based about the premise of stochastic optimization problems are transformable to deterministic problems controlling the probability of route failure events occur. Nevertheless, this artifitial control might result in bad routing decissions.

2.2.1.2. *Stochastic programming with resources*

In SPRs, the aim is to determine a first stage solution that minimizes the expected cost of the second stage solution: this cost is made up of the cost of the first stage solution, plus the expected net cost of recourse. SPRs are typically more difficult to solve than are CCPs, but their objective function is more meaningful [**19**].

Yang [**34**] proposse two heuristic methods to solve the problem and Laporte et. al. [**23**] and Gendrau et. al. [**18**] proposse a *L-shape* method to find optimal solutions, i.e. an branch-and-cut algorithm adjusted for the stochastic approach. Below we present the model formuled by [**15**] based on Laporte model although allowing proactive replenishments of the vehicle in a single route, supported on Yang's affirmation of it is more efficient than multiple routes.

Let $T(\hat{x}, D) = \sum_i^n \sum_j^n d_{ij}x_{ij}$ be the costo of the routing solution where $\hat{x} = \{x_{ij}\}, i \in V, j \in V$ is the vector of routing decisions, $\hat{x}_{ij} = 1$ if the vehicle visits directly the node i from j node and 0 otherwise, D is a vector of the customer demands disclosed one ath a time when the vehicle arrive at customer location. Both $\hat{x}$ and D are random variables

$$\min_{\hat{x}} E_D[T(\hat{x}, D)] \tag{2.2.4}$$

$$\text{subject to} \sum_{i=0}^{n} \hat{x}_{ij} \geq 1, \forall i \in V \tag{2.2.5}$$

$$\sum_{j=0}^{n} \hat{x}_{ij} \geq 1, \forall j \in V \tag{2.2.6}$$

$$\sum_{i \in S} \sum_{j \notin S} \hat{x}_{ij} \geq 1, S \subseteq 1, \ldots, n; |S| \geq 2 \tag{2.2.7}$$

$$\hat{x}_{ij} \in 0, 1, i, j \forall i, j \in V \tag{2.2.8}$$

$T(\hat{x}, D)$ can be divided in two parts, in the first part we have the term cx denoting the cost (distance) of the *a priori* sequence represented by x, and at the second part $\mathcal{Q}(x, D)$, would be the cost of recourse given x and a realization of D. $\mathcal{Q}(x, D)$ represent the cost of return trips incurred by route failures, minus some resulting savings. $T(\hat{x}, D) = cx + \mathcal{Q}(x, D)$ where x represents an TSP route and $\hat{x}$ is the binary routing

vector which includes all the resources decissions. For keeping $\hat{x}$ as binary is assumed that the probability of a node demand to be greater than capacity of an vehicle is zero as well as the probability that a vehicle, upon failure, returning to a node to complete its delivery after visit the depot is also zero.

Setting the expectation $\mathcal{Q}(x) = E_D[\mathcal{Q}(x, D)]$ the objective function becomes in

$$\min_x cx + \mathcal{Q}(x) \min_x cx + E_D[\mathcal{Q}(x, D)] \tag{2.2.9}$$

In the standard modeling of the Two-Stage Stochastic Linear Programs, in the secon stage the suctomers deliveries are represented as

$$\mathcal{Q}(x, D) = \min_y \{cy | Wy = h(D) - T(D)x, y \in Y\} \tag{2.2.10}$$

Where y is the binary vector representing the recourse initiated trips to the depot, $T(D)$ represents the deliveries made by the x vector given D and $h(D)$ is the demand realization fro D which has to be met (delivered) either by x or the resource y. Below we show the model used by **[23]** for the L-shape method

$$\min_x \{cx + \mathcal{Q}(x)\} \tag{2.2.11}$$

$$\text{subject to} \sum_{i=0}^{n} x_{ij} = 1, \forall i \in V \tag{2.2.12}$$

$$\sum_{j=0}^{n} x_{ij} = 1, \forall j \in V \tag{2.2.13}$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1, S \subseteq 1, \ldots, n; |S| \geq 2 \tag{2.2.14}$$

$$\hat{x}_{ij} \in 0, 1, i, j \forall i, j \in V \tag{2.2.15}$$

A extended review of models presented above and others for VRPSD is presented by **[15]** and **[14]**

2.2.2. Stochastic Dynamic Programming

Stochastic Dynamic Programming (SDP) provide a frame-work where decisions are made in an finite number of stages under uncertainty, in these problems there are a set of states S, decisions variables U and the uncertainty is represented by a set of random variables W, the dynamic system is of the form

$$f : U \times W \times S \to S$$

or

$$x_{k+1} = f_k(x_k, u_k, w_k), k = 0, 1, \ldots, N-1 \tag{2.2.16}$$

where $x_k \in S_k$ is the state of system at $k-th$ time and summarizes past information, u_k is the control or decision variable to be selected in a given nonempty subset $U(x_k) \subset U$ which depends on the current state x_k, i.e. $u_k \in U_k(x_k) \forall x_k \in S_k$, $w_k \in W$ is a random parameter whose value is disclosed at time k it's characterized by a probability distribution $P_k(\cdot|x_k, u_k)$ that may depend on x_k and u_k but not on prior values of the random variable $w_{k-1}, \ldots, w_0$. N is the horizon or number of times that control is applied

The objective is to minimize a cost function $g_k(x_k, u_k, w_k)$ of the form

$$g_k : S \times U \times W \to \mathbb{R}$$

The cost function is assumed additive, i.e. the cost incurred accumulates over time.

$$g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)$$

Nevertheless, given uncertainty the cost is a random variable, so we fomulate the problem as an optimization of the expected cost

$$E\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) \right\} \tag{2.2.17}$$

We define *admissible policies* $\pi \in \Pi$, where Π is the set of all admissible policies, as a sequence of functions $\mu_k : x_k \to u_k$, in which μ_k maps states x_k to controls $u_k = \mu_k(x_k)$ such that $\mu_k(x_k) \in U_k(x_k) \forall x_k \in S_k$.

$$\pi = \mu_0, \ldots, \mu_{N-1}$$

Given a policy π and a initial state x_0 the equation 2.2.16 is becoming in

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), k = 0, 1, \ldots, N-1 \tag{2.2.18}$$

making x_k and w_k random variables with probability distributions defined. Hence, is well defined the expected cost function g_k 2.2.17, with $k = 0, 1, \ldots, N$

$$J_\pi(x_0) = E_{w_k}\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\} \tag{2.2.19}$$

An optima policy π^* for a given initial state x_0 is one that minimizes the cost $J(x_0)$, i.e

$$J_{\pi^*}(x_0) = J^*(x_0) = \min_{\pi \in \Pi} J_\pi(x_o)$$

The SDP model applied to VRPSD is formulated below in the section 2.2.3

2.2.2.1. *Finite-Stage Models*

Suposse that the states are the integers, and let A, a finite set, be the set of all possible actions. $R(i, a)$ is the reward in ith-state given that $a \in A$ action was chosen, and the next state is j with probability $P_ij(a)$. Let $V_N(i)$ denote the maximum expected return for an N-stage problem that starts in state i.

When $N = 1$ we have:

$$V_1(i) = \max_{a \in A} R(i, a) \tag{2.2.20}$$

Considering a N-stage problem that starts in i and has $N - 1$ time periods to go. We can assess the expected return given initially we choose action a:

$$R(i, a) + \sum_j P_{ij}(a) V_{N-1}(j)$$

then,

$$V_N(i) = \max_a [R(i, a) \sum_j P_{ij}(a) V_{N-1}(j)] \tag{2.2.21}$$

2.2.3. Stochastic Dynamic Programming approach for VRPSD

VRPSD is modeled in the framework of SDP as a stochastic shortest path problem, it's a markov decission process (MDP) where is necesary make decisions under situations where outcomes are partly random reaching an absorbing cost-free termination state in a random number of stages.

The problem formulation is presented below, based in the Novoa [**26**] and Secomandi [**27**] notation as a Markov decision model.

The objective of the problem is to find a routing policy such that customer demand is satisfied and minimized expected transportation costs (distance), this policy may order returns to the depot before the vehicle capacity runs out.

2.2.3.1. *Types of policies*

Secomandi [**29**] classify the routing policies in three groups, static, dynamics and mixed

Static: Static policies describe a sequence τ of customers to be visited in that order for the vehicle.

Dynamic: Dynamic policies provide a policy π that given the current state of the system prescribe witch location should be visited next.

Mixed: Mixed policies combine elements of both static and dynamics policies.

Mixed policies not only following a sequence τ of customers but also prescribe decisions dependent of the state that allow proactive replenishments. We ilustrade in the figure 2.2.1 a static policy (left) where a reactive replenishments or resources action carry out when the customer demand is greater than vehicle capacity and it is forced to return to the depot for restocking, while on the right is represented a dynamic policy in which the vehicle can do proactive replenishments going to the station even when the vehicle capacity is not empty.

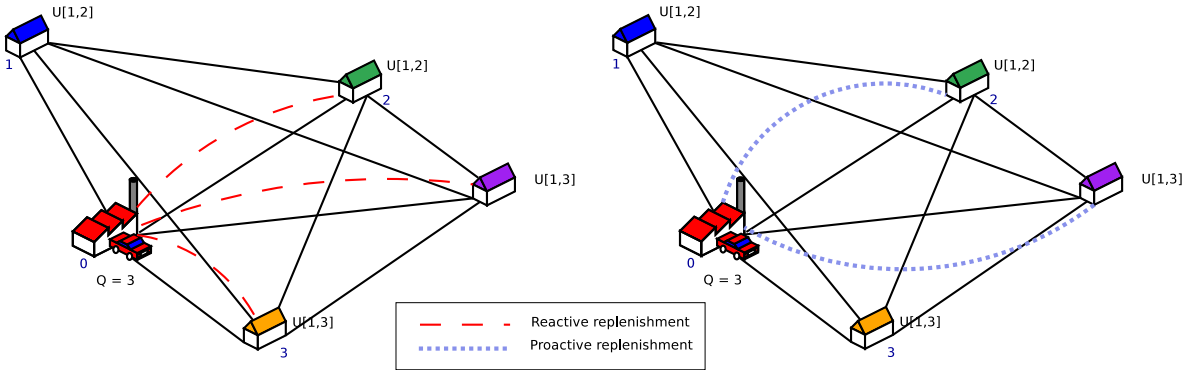

FIGURE 2.2.1. static and mixed routing policies

To represent the system state at stage k, the vector x_k is defined as $x_k = (l, q_l, r_1, \ldots, r_n)$ of size $n + 2$, where $l \in \{0, 1, \ldots, n\}$, is the current location of the vehicle and $q_l \leq Q$ is its capacity available *after* delivery to customer l; the elements r_i represents the remaining demand to satisfy to the costumer i. A unknown demand is denoted as -, if customer i is visited and its demand has been satisfied completely r_i will take the value 0, otherwise, take any value between 1 and R. The initial state of the system x_0 is

$(0, Q, -, -, \ldots, -)$ and the final state occurs when the vehicle returns to the depot after serving the demands of customers, represented as $(0, Q, 0, 0, \ldots, 0)$. Thus, the number of states in the system is $O(nQR^n)$

Let N be a random variable representing the number of stages or transitions from initial state to the end, the vector $\pi = \mu_0, \mu_1, \ldots, \mu_{N-1}$ is the policy or sequence of functions to optimize, where μ_k is a function that associates a decision or control $u_k = \mu_k(x_k)$ for each state, $u_k \in U_k(x_k)$ and $U_k(x_k) = \{\{m \in \{1, \ldots, n\}\} | r_m \neq 0\} \cup 0\} \times \{a : a \in \{0, 1\}\}$. Control u_k is represented as ordered pairs (m, a), m is any costumer not yet served, m is 0 when all demands have been satisfied and the system enters its completion stage, a is 0 if the vehicle visits customers directly and 1 if the vehicle stops first at the depot for resupply.

Given a state $x_k = (l, q_l, r_1, \ldots, r_m, \ldots, r_n)$ and a control u_k in which is decided to visit the node m at the next stage, the random variable D_m is realized ($r_m = D_m$ if r_m is unknowed otherwise $r_m \neq ?$) and the remaining demand of the customer m change to r'_m as soon as the capacity of vehicle becomes in q_m, where

$$q_m = \begin{cases} max(0, q_l - r_m), & \text{whether } u_k(m, 0) = \mu_k(x_k) \\ q_l + Q - r_m, & \text{whether } u_k(m, 1) = \mu_k(x_k) \end{cases} \tag{2.2.22}$$

and

$$r'_m = \begin{cases} min(0, r_m - q_l), & \text{whether } u_k(m, 0) = \mu_k(x_k) \\ 0, & \text{whether } u_k(m, 1) = \mu_k(x_k) \end{cases} \tag{2.2.23}$$

so the system goes to state $x_{k+1} = (m, q_m, r_1, \ldots, r'_m, \ldots, r_n)$. The transition between states is graphically represented as:

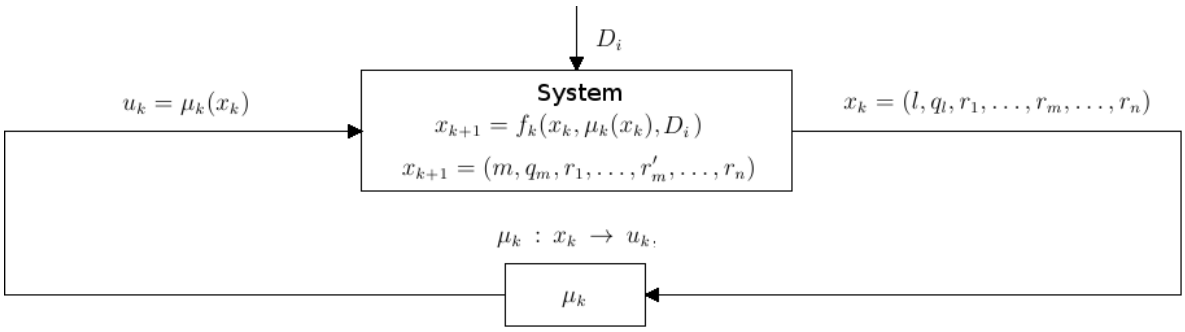

FIGURE 2.2.2. Stochastic Dynamic System for VRPSD

Incurring in a transition cost $g(x_k, u_k, x_{k+1})$

$$g(x_k, \mu_k(x_k), x_{k+1}) = \begin{cases} d(l,m), & \text{whether } u_k(m,0) = \mu_k(x_k) \\ d(l,0) + d(0,m), & \text{whether } u_k(m,1) = \mu_k(x_k) \end{cases} \quad (2.2.24)$$

The objective of the problem is to find a policy π that minimizes the cost of transport J_N^π (2.2.25) in the N-stages or the expected cost to complete a given initial state. The optimal cost of transport in the N-stage x is $J_N^*(x) = min_{\pi \in \Pi} J_N^\pi(x)$, where Π is the set of admissible policies.

$$J_N^\pi(x_0) = E\left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), x_{k+1}) \right\} \quad (2.2.25)$$

If $J_N^*(x)$ is known for all stages, the optimal control u_k^* at each stage is to find the minimum of the following equation (2.2.26):

$$u_k^* = \mu_k^*(x) = \arg \min_{u_k \in U_k(x_k)} g(x_k, u_k, x_{k+1}) + \sum_{x_{k+1} \in S} p_{x_k x_{k+1}}(u_k) J_N^*(x_{k+1}) | x_k = x, \forall x \in S \quad (2.2.26)$$

The problem is that $J_N^*(x)$ is unknown and its calculation is a computationally intractable problem given the size of state space. Secomandi **[27]** point out that compute an optimal policy becomes quickly intractable when n grows beyond 10. The chapter 3 deal with approximate this function through of dynamic-programming method efficiently computable.

2.3. Summary

The VRPSD have been studied by more than 20 years, with important progress in 90's and 00's and wide areas of application in logistics following the conclusions of **[15]** the mos promessing approach is modeling the problem as a Markov decision process, Therefore, a stochastic programming model is selected, it's a methodology for sequential decisions making under uncertainty based on dynamic system, the main idea is to use a approximate a function J to make decisions in complex dynamic systems allowing deal with instances considered intractable for their size. In the next section the dynamic programming solution is addressed.

CHAPTER 3

Stochastic Dynamic programming solution

3.1. Dynamic approach for VRPSD

Dynamic programming is based on principle of optimality formuled by Bellman [**1**]

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

Following to Bertsekas [**3**] the principle of optimality point out that an optimal policy can be constructed backward, first finding an optimal policy for the *tail subproblem* involving the last stage, then extending the optimal policy to the problem regard with the last two stages, and continuing until cover the whole problem in the first stage, hence a optimal policy is constructed for the entire problem.

$$J^*(x_0) = \min_{u_k^* \in U_k(x_k)} E_{w_k}\left\{ g_k(x_k, u_k^*, w_k) + J_{k+1}^*(f(x_k, u_k^*, w_k)) \right\}, \quad \forall k = 0, 1, \ldots, N-1 \quad (3.1.1)$$

However, the exact assesing of J^* is computationally imposible given the size of the state space. Therefore, is necessary approximate it function for generate good, but not necessarily optimal policies.

Let $\tilde{J}_k$ be an approximation of J_k^*, then a control can $\tilde{u}_k$ be assesed

$$\tilde{u}_k = \tilde{\mu}_k(x_k) = \arg\min_{u \in U_k(x_k)} \left\{ g(x_k, u_k, x_{k+1}) + \sum_{x_{k+1} \in S} p_{x_k x_{k+1}} \right\} \quad (3.1.2)$$

The following section 3.1.1 discuss the computation of $\tilde{J}_k$

3.1.1. Expected distance

The expected distance $\tilde{J}$ or *cost-to-go* is computed based on the algorithm propossed by Secomandi [**27**]. We implemented the algorithm Γ represented below 1 to compute expected distance in $O(nRQ)$ time and $O(nQ)$ space.

input : tour $\tau_{1\times n+2}$, $d_{n+1\times n+1}$ distance, x state
output: E expected distance of an apriori solution τ (base sequence)
$l = x_1$;
$q_l = x_2$;
if $l = n$ **then** is the last customer on τ
  $E = d(\tau(l+1)+1, 1)$;
end
else
  if *l=0* **then** is the first node *depot* in the tour τ
    E = $\Gamma(\tau, l+1, q_l)$;
  end
  else
    $E_0 = d(\tau(l+1), \tau(l)) + \sum_{j=0}^{\min\{q_l, \bar{D}_{\tau(l+1)}\}} \Gamma(\tau, l+1, q_l - j) *$
    $p^{\tau(l+1)}(j) + \sum_{j=q_l+1}^{\bar{D}_{\tau(l+1)}} 2 * d(0, \tau(l+1)) + \Gamma(\tau, l+1, Q + q_l - j)) * p^{\tau(l+1)}(j)$;
    $E_1 = d(0, \tau(l)) + d(0, \tau(l+1)) + \sum_{j=0}^{\bar{D}(\tau(l+1))} \Gamma(\tau, l+1, Q - j) * p^{\tau(l+1)}(j)$;
    $E = \min\{E_0, E_1\}$;
  end
end

Algorithm 1: Expected distance algorithm $E = \Gamma(\tau, l, q_i)$

This algorithm was validated comparing the outcomes with the expected distance computed by an exhaustive algorithm applied to small instances. In addition, larger instances was benchmarked using monte carlo simulation to asses expected distance.

If the vehicle capacity is depleted, the methods used to compute the expected distance take into account that the vehicle can go to the depot for proactive restocking with less cost than to visit the customer first which makes the route fail, as we show in 3.1.1.

THEOREM 3.1.1. *if $q_l = 0$ then the vehicle must go first to the depot for replenishment and later visit the next customer on the route, it is better than it visit the next customer to know its demand and go to the depot for replanishment after.*

PROOF. Given the current state $x_k = (l, 0, r_1, \ldots, r_n)$ where the vehicle capacity is depleted, i.e. $q_l = 0$, assume that the next customer to be visited on the route is l'. If the customer is visited first, then the vehicle must go to depot for replenishment and go back to l' to satisfy its demand, so the distance is $\delta(l, l') + 2\delta(l', 0)$. Otherwise, if a proactive restocking of the vehicle is performed then the total distance is $\delta(l, 0) + \delta(0, l')$. Assume $\delta(l, 0) + \delta(0, l') \leq \delta(l, l') + 2\delta(l', 0)$ then by triangular inequality $\delta(l, 0) \leq \delta(l, l') + \delta(l', 0)$ proving 3.1.1 □

LEMMA 3.1.2. *The theorem 3.1.1 can be generalized for all $q_l \geq D'_i$ where D'_i is the demand of the customer i who is the next on the tour.*

3.2. Policy iteration

The policy iteration algorithm generates a sequence of stationary policies, each with improved cost over the preceding one.

(1) Initialization
(2) Policy evaluation
(3) Policy improvement

Step 1: Guess an initial stationary policy π_0
Step 2: Given the stationary policy π_k compute the corresponding cost function J_{π_k}
Step 3: Obtain a new stationary policy π_{k+1}

Algorithm 2: Policy iteration algorithm

3.3. Neuro-Dynamic programming

Neuro Dynamic programming has been designed to deal with dynamic programming problems where the number of states is too large or is unknowed completely.

[?]

3.4. Approximate policy iteration

3.5. Optimistic Approximate policy iteration

3.6. Rollout policy

[?]

3.6.1. Rollout algorithm

RA require a base policy The rollout algorithm build a policy π starting from any given state and following an *a priori* solution to VRPSD

For changing τ after the first state, when there are a transition of x_0 state to x_l, is only shifted the segment of τ that follow to l, maintaining in this section the customers still no visited or with demand different to 0.

The shift (cyclic heuristi)c is perfomed to partial tour not included at the policy.

3.6.1.1. *Defining initial policy*

cyclic heuristic propossed by Bertsimas 92 and improved by himsel 95 and later used by a lot of authors because it is simple

cyclic heuristic $\mathcal{C}$ builds the cyclic tour that start at l and follow τ cyclically, then $\tau_l^{\mathcal{C}} = (l, l+1, \ldots, n, 1, \ldots, l-1, 0)$ represent an apriori policy $\pi^{\mathcal{C}}$

input : tour $\tau_{1\times n+2}$, state $x_{1\times n+2}$
output: π policy
$\bar{\tau} = \tau$;
$i = 1$;
while $x \neq x_f$ **do**
  $\tau = \bar{\tau}$;
  for $j \in SN$ **do**
    if *i=1* **then** is the first node in the tour τ, i.e. l=0
      $\tilde{J} = \Gamma(\tau, 0, q_l)$;
      $a_{min} = 0$;
    end
    else
      $J^0 = g^0(i, \tau_{i+1}, x)$;
      $J^1 = g^1(i, \tau_{i+1}, x)$;
      $\tilde{J} = \min\{J^0, J^1\}$;
      $a = \arg\min\{J^0, J^1\} - 1$;
    end
    if $\tilde{J}_{min} > \tilde{J}$ **then**
      $\tilde{J}_{min} = \tilde{J}$;
      $a_{min} = a$;
      $\bar{\tau} = \tau$;
      $\tau = sh(\tau, i)$;
    end
  end
  $l = \bar{\tau}_{i+1}$;
  $\pi \leftarrow u = (l, a_{min})$;
  $x = \Upsilon(x, u)$;
  if $r_l = 0$ **then**
    $SN = SN - l$;
  end
  $i = i + 1$;
end

Algorithm 3: Rollout algorithm

where SN is the set of nodes that still need to be visited, i.e. $\forall l \in N, D_l > 0$, x_f is the final state, where the vehicle comeback to depot and each customer was visited and its demand is 0, $x = (0, Q, 0, \ldots, 0)$, $\bar{\tau}$ is the minimum τ selected in each algorithm iteration. $q_l = x_2$, $l = x_1$ and $r_l = x_{l+2}$
When $l = 0$ or $i = 1$ in the algorithm, $q_l = Q$ $sh(\tau, i)$ shift sub τ vector from position i to the final position.

The $g^a(l, m, x)$ function is the expected distance from l to m given the state x, where $a = 1$ if earlier replanishment is specified or $a = 0$ in otherwise. The function is described below.

$$g^0(l,m,x) = d(\tau_l, m) + \sum_{k=0}^{q_l} p_m(k)\Gamma(\tau, l{+}1, q_l{-}k) + \sum_{k=q_l+1}^{K_m} 2d(0,m)p_m(k)\Gamma(\tau, l{+}1, q_l{+}Q{-}k) \tag{3.6.1}$$

$$g^1(l,m,x) = d(0,\tau_l) + d(0,m) + \sum_{k=0}^{K_m} p_m(k)\Gamma(\tau, l+1, Q-k) \tag{3.6.2}$$

$x_l = \Upsilon(x, u)$ represent the transition of the state x to state x_l given that the control u is realized.

CHAPTER 4

Hybrid Evolutionary approach

4.1. Hybrid Genetic Algorithm

A evolutionary algorithm is a technique bio-inspired in the evolution of some structures, it was proposed in first time by John Holland in the 60's, this algorithm seeks to evolve a population of individuals that represent solutions to the problem through genetic operators such as crossover, mutations and selection, the population generated in each iteration is evaluated and then the individuals that had a better fitness are chosen for the next generation with more probability.

A hybrid evolutionary algorithm or hybrid genetic algorithm are very popular techniques that offers practical advantages to deal with complex and hardly optimization problems. (Grosan, 2007) present a review of hybrid genetic architectures frecuently used.

"Evolutionary algorithm behavior is determined by the exploitation (improve solution - local search) and exploration (avoid local optimum extending the search space) relationship kept throughout the run"

hibridization is used for: Improve the performance of the evolutionary algorithms Improve quality of solutions obtained by evolutionary algorithm Incorporate evolutionary algorithm as part of a large system

hybridization can be performed using prior knowledge, heuristics, local search, other techniques

local search is knowed as memetic algorithm

local search can be incorporated in the initial population or among the offsprings

4.1.1. A basic genetic algorithm for vehicle routing problem with stochastic demands

A example of a basic GA used to find solutions to VRPSD is showed in general form in figure 4.1.1. First a initial population is selected, for each individual in the population assess fitness function, since individuals with better fitness function has more chance to reproduce then croossover operators are applied to selected individuals, also mutations can be performed in this stages, generating a offspring with new individuals. Probably,

they will be added to new population whether have a fitness competitive respect to the population. So, a selection of individuals is applied to conform the new population, individuals with better fitness have more probability to get it. Until stopping criteriums are resolve, reproduction and selection is applied repeatedly over the population.

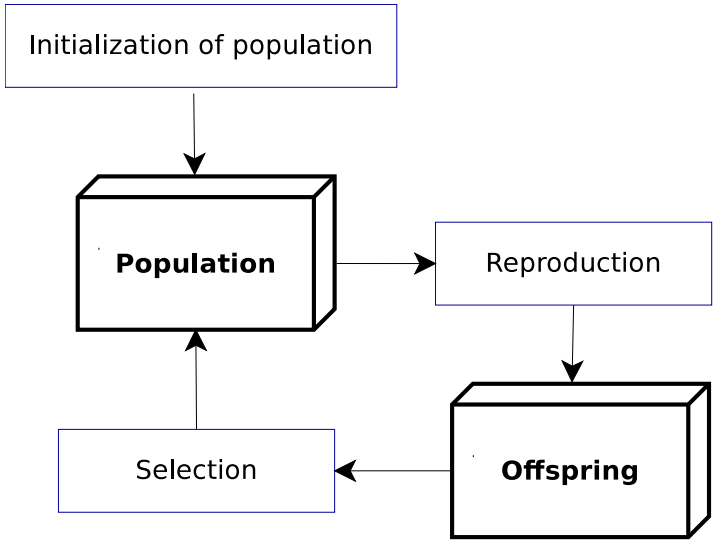

FIGURE 4.1.1. Basic genetic algorithm

Often hybridazation is included in inicialization and reproduction operators.

What is The fitness function?

The crossover operator consist in selecting two different individuals I_1 and I_2 with probability dependently of their fitness value respectively, in contrast a random cut point γ is selected uniformly $\gamma \in [1, n]$, then a new individual raise of concatenate the part I_1 sequence of 1 to the cut point with the part of sequence I_2 of cut point to n, at some times this process can be inverted, e.g. $I_1 = \{2, 3, 4, 5, 1\}$, $I_2 = \{2, 3, 4, 1, 5\}$, then $I_1 \times I_2 = \{2, 3, 4, 1, 5\}$. When a element in the crossover process can be generate a unfeasible sequence is corrected.

The mutation can happened with probability $\alpha = 0.2$ in the experiments presented in this document, this operator swaps two elements of an tour selected randomly.

A classic genetic algorithm does not yield competitive results, due to simple GA does not exploit problem knowing to produce high quality solutions. To be effective, we combine local search methods.

We represents individuals as a policy tour

Size of population:

Deprecated: Size of population is $\lfloor \alpha \log n \rfloor$ where α is a tuneable parameter which should be fixed depending of computational resources

Used: Size of population is $\lfloor \max\{n + \alpha(n\Delta_{E'}), n/4\} \rfloor$ where α is a tuneable parameter which should be fixed depending of computational resources, it reward a offspring that improve the quality of the solutions regard to theirs parents increasing the population size, otherwise, when the quality of the solutions decrease the size of population is punished.

4.1.2. Genetic operators

4.1.2.1. *Crossover - EAX*

Edge assembly crossover

Mutation

Swap

local search

Performance of the genetic algorithm

This section describes the algorithm performance for a specific problem instance with 10 customers and a factor $f' = 1$, the customers' location and demand value (known when the vehicle arrive to the customer location) are showed in the figure **??**.

The genetic algorithm outcome for the instance problem described above is presented in the figure **??**, the red line describes the route, the segmented black line represents returns to the depot for replenishment, this tour was selected as the best solution found in 100 iterations of the GA.

The figure **??** shows the improving of the objective function found, the chart shows the lesser total cost history, when a better value is found the line fall, therefore we can see the improving is given among the iteration 25 and 40, after this iterations the algorithm converges and it's not found a better solution.

CHAPTER 5

Experiments and Numerical Results

5.1. Instances of VRPSD

To compare results we are used instances of two diferents sources, the first set of instances was generated following a procedure similar to the used in Secomandi [**29**], the another set of instances is the used by Novoa [**26**], Both instance sets are generated using the procedure described on the next section.

5.1.1. Instance generation

The set of instances contain 45 different instances resulting from combine three number of customers $n \in \{5, 10, 20\}$, three vehicle capacities given for f' factor and five different assignments for customer locations and demand distribution for each one, the assignments result from changing the random seeds.

The customers' demands are both discrete and distributed uniformly in this posible sets $U[1, 5]$, $U[3, 9]$, $U[6, 12]$, in each instance, each customer is assigned to any of the three groups with equal probability. Customers' locations are random points in $[0, 1]^2$, with the depot fixed at $(0, 0)$.

The filling rate f is an index of the total expected demand relative to vehicle capacity.

$$f = \sum_{i=1}^{n} \frac{E[D_i]}{mQ} \tag{5.1.1}$$

where $E[D_i]$ is the expected demand of customer i, m is the number of available vehicles, when $m = 1$ f can represent approximately the expected number of replenishment needed to serve all customers' demands. It follows that, a priori, in all instances $E[D_i] = (3 + 6 + 9)/3 = 6$, for any customer i, and $Q = 6n/f$. $f' = f - 1$ is the expected number of route failure in a given instance. Hence, we is defined for this factor $f' \in \{1.0, 1.5, 2.0\}$, following the same factors used by Secomandi [**29**].

The 160 instances used by Novoa [**26**] are composed by 70 small size instances (5 to 20 vertex), 60 medium size (30 to 60 vertex) and 30 instances of large size whose number of vertex is greter than 100.

TABLE 5.1.1. Vehicle capacity for each factor

f'	n		
	5	10	15
1.0	15	30	45
1.5	12	24	36
2.0	10	20	30

	demand values									
n	4	5	7	8	9	15	17	29	33	Total
5	2	4	1	1	8	2	3			21
8		4	2		8		5			19
20					5	5	10	5	5	30
30						5	5	5	5	20
40						5	5	5	5	20
60						5	5	5	5	20
100						5	5	5	5	20
150						5	5			10

TABLE 5.1.2. Instances characterization

5.2. Assess the expected distance for a given policy

straightforward policy is better than cyclic policy at each one instance tested 5.2.1

5.2.1. Expected distance algorithm

Validation of backward expected distance algorithm comparing with the average expected distance of a tour τ, where τ is assumed as a sorted permutation of the nodes, i.e. $\tau = (1, \ldots, n, 0)$. The average expected distance is computed using monte carlo simulation under policy π following τ and considering early replenishments defined by the policy.

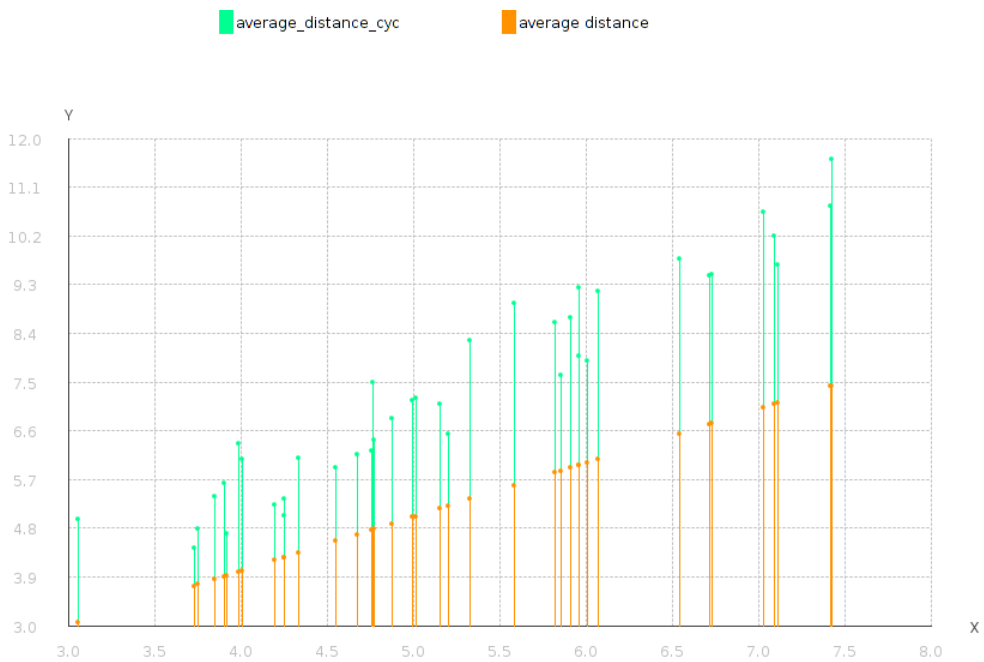

FIGURE 5.2.1. average distance straightforward tour vs. average distance cyclic tour

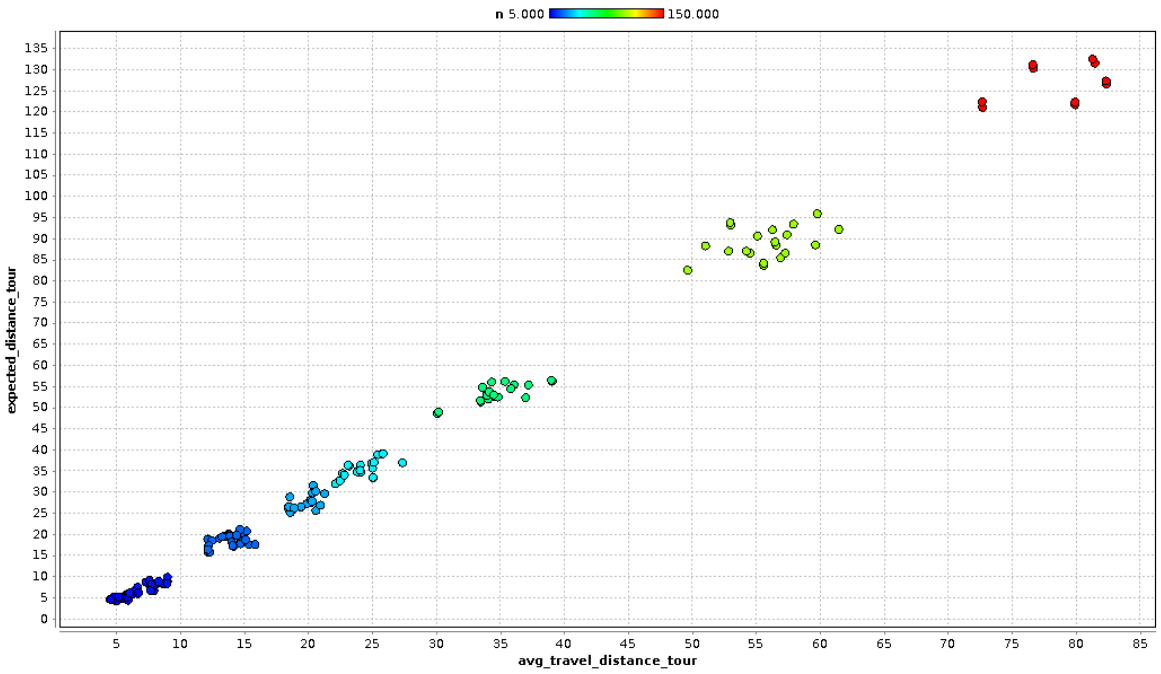

FIGURE 5.2.2. average distance tour vs. expected distance tour

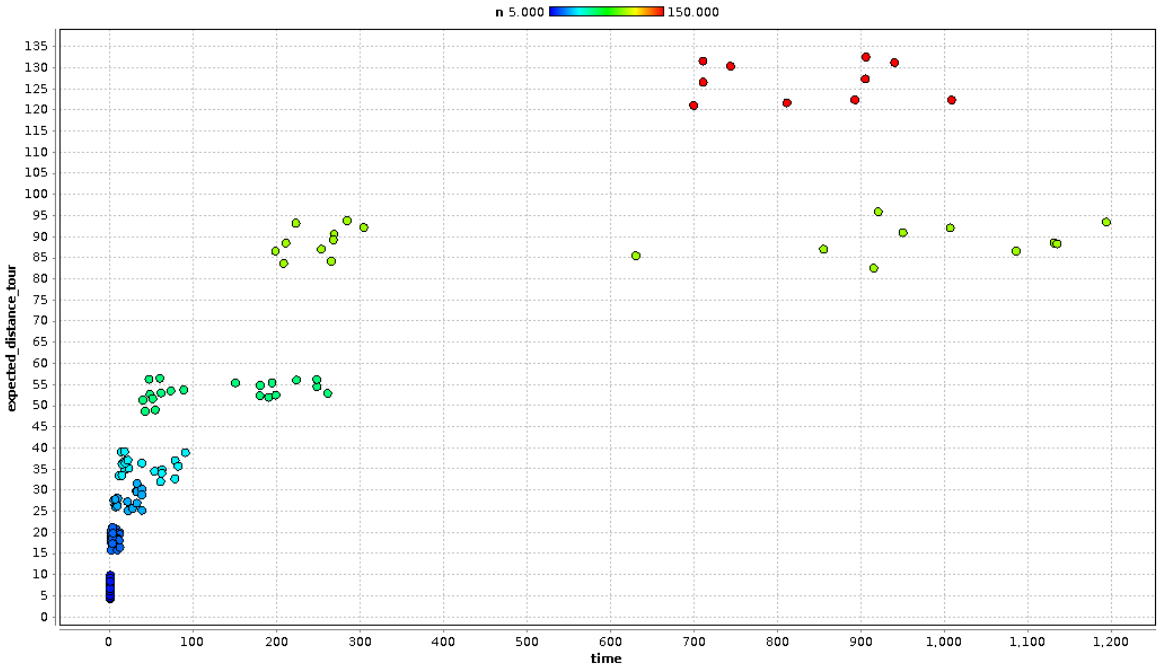

FIGURE 5.2.3. execution algorithm time vs. expected distance tour

CHAPTER 6

Conclusions

6.1. Conclusions

Although the hybridization approach increases the computational time cost, this effort is rewarded in many cases given that the quality of the solutions is improved than if we apply the rollout algorithm alone.

Hence, since the local searchis performed every time a new solution is generated.

Often memetic algorithms employ long runnig times because local search is the most time-consuming component. For this reason, we partialy apply the rollout algorithm in large instances to reduce the time consumption to compute the solution.

Although the genetic operators attempt to explore by extending the search space to avoid local optimum. This operators by themselves do not improve the solutions. So, it is necessary to focus on local search to improve the quality solutions and the convergence.

To evaluate expected distance consume a lot of computational time, so an efficient approximation improves the algorithm performance.

Given that local search is performed every time a new solution si generated often memetic algorithms employ long runnig times.

Local search is the most time-consuming component.

Evaluate expected distance consume a lot of execution time, efficiently approximation improve the algorithm performance.

Focus local search improve the quality solutions and the convergency.

6.2. Perspectives

Bibliography

[1] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc*, 60(6):503–515, 1954.

[2] Dimitri Bertsekas. Dynamic programming and stochastic control. *Journal of the American Statical association*, 74:510–511, 1979.

[3] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[4] Dimitri P. Bertsekas. *Differential Training Of Rollout Policies*. 1997.

[5] Dimitris J. Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585, May 1992.

[6] Dimitris J. Bertsimas and Garrett van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4):601–615, August 1991. ArticleType: primary_article / Full publication date: Jul. - Aug., 1991 / Copyright Âĺ 1991 INFORMS.

[7] Leonora Bianchi, Mauro Birattari, Marco Chiarandini, Max Manfrin, Monaldo Mastrolilli, Luis Paquete, Olivia Rossi-Doria, and Tommaso Schiavinotto. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms*, 5(1):91–110, April 2006.

[8] Homem-De-Mello T Chepuri K. Solving the vehicle routing problem with stochastic demands using the cross entropy method. *Annals of Operations Research*, 55:153–181, 2005.

[9] Bernard K.-S. Cheung, K.L. Choy, Chung-Lun Li, Wenzhong Shi, and Jian Tang. Dynamic routing model and solution methods for fleet management with mobile technologies. *International Journal of Production Economics*, 113(2):694–705, June 2008.

[10] Christian H. Christiansen and Jens Lysgaard. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35(6):773–781, November 2007.

[11] G. B. Dantzing and J. H. Ramser. The truck dispatching problem, Oct 1959.

[12] Maged Dessouky, Fernando Ordonez, Hongzhong Jia, and Zhihong Shen. Rapid distribution of medical supplies. In *Patient Flow: Reducing Delay in Healthcare Delivery*, pages 309–338. 2006.

[13] M. Dror, M. Ball, and B. Golden. A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4(1):1–23, December 1985.

[14] Moshe Dror. Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution. *European Journal of Operational Research*, 64(3):432–441, 1993.

[15] Moshe Dror. Vehicle routing with stochastic demands: Models & computational methods. In *Modeling Uncertainty, International Series in Operations Research & Management Science*. Springer New York, 2005.

[16] Jianhua Fan, Xiufeng Wang, and Hongyun Ning. A multiple vehicles routing problem algorithm with stochastic demand. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 1, pages 1688–1692, 2006.

[17] Noah Gans and Garrett van Ryzin. Dynamic vehicle dispatching: Optimal heavy traffic performance and practical insights. *Operations Research*, 47(5):pp. 675–692, 1999.

[18] Michel Gendreau, Gilbert Laporte, and Rene Seguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *TRANSPORTATION SCIENCE*, 29(2):143–155, May 1995.

[19] Michel Gendreau, Gilbert Laporte, and Ren Sguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[20] Ali Haghani and Soojung Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11):2959–2986, November 2005.

[21] Ivana Cavar Ton?i Cari? Hrvoje Markovic. Using data mining to forecast uncertain demands in stochastic vehicle routing problem. In *13th International Symposium on Elecronics in Transport (ISEP), Slovenia*, 2005.

[22] Raja Jothi and Balaji Raghavachari. Approximating the k-traveling repairman problem with repairtimes. *Journal of Discrete Algorithms*, 5(2):293–303, June 2007.

[23] Gilbert Laporte, FranÇois V. Louveaux, and Luc Van Hamme. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, May 2002.

[24] Jorge E. Mendoza, Bruno Castanier, Christelle Guéret, Andrés L. Medaglia, and Nubia Velasco. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11):1886–1898, November 2010.

[25] Rodrigo Moretti. Adaptive granular local search heuristic for a dynamic vehicle routing problem. *Computers & Operations Research*, 36:2955–2968, 2009.

[26] Clara Novoa and Robert Storer. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196(2):509–515, July 2009.

[27] N. Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, pages 796–802, 2001.

[28] Nicola Secomandi. *Exact and Heuristic Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands*. PhD thesis, Deparment of Decision and Information Sciences, University of Houston, Houston, TX, 1998.

[29] Nicola Secomandi. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 27(11-12):1201–1225, September 2000.

[30] Alan Slater. Specification for a dynamic vehicle routing and scheduling system. *International Journal of Transport Management*, 1(1):29–40, February 2002.

[31] Martin A. Wortman Tapas K. Das. Analysis of asymmetric patrolling repairman systems. *European Journal of Operational Research*, 64:45–60, 1993.

[32] Eldon Y. Li Timon C. Du and Defrose Chouc. Dynamic vehicle routing for online b2c delivery. *The international journal of management science*, 33:33–45, 2004.

[33] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2001.

[34] Wen-Huei Yang, Kamlesh Mathur, and Ronald H. Ballou. Stochastic vehicle routing problem with restocking. *TRANSPORTATION SCIENCE*, 34(1):99–112, February 2000.