

University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Algorithm portfolios for logistics optimization considering stochastic demands and mobility allowance

N Shukla

University of Warwick, nshukla@uow.edu.au

A K. Choudhary

University Of Sheffield

P Prakash

Irish Centre for Manufacturing Research

K J. Fernandes

University of York

M K. Tiwari

Indian Institute of Technology - Kharagpur

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Algorithm portfolios for logistics optimization considering stochastic demands and mobility allowance

Abstract

The vehicle routing problem with stochastic demand (VRPSD) is a well known NP-hard problem. The uncharacteristic behaviour associated with the problem enhances the computational efforts required to obtain a feasible and near-optimal solution. This paper proposes an algorithm portfolio methodology based on evolutionary algorithms, which takes into account the stochastic nature of customer demand to solve this computationally complex problem. These problems are well known to have computationally complex objective functions, which make their solutions hard to find, particularly when problem instances of large dimensions are considered. Of particular importance in such situations is the timeliness of the solution. For example, Apple was forced to delay their shipments of iPads internationally due to unprecedented demand and issues with their delivery systems in Samsung Electronics and Seiko Epson. Such examples illustrate the importance of stochastic customer demands and the timing of delivery. Moreover, most of the evolutionary algorithms, known for providing computationally efficient solutions, are unable to always provide optimal or near optimal solutions to all the VRPSD instances within allocated time interval. This is due to the characteristic variations in the computational time taken by evolutionary algorithms for same or varying size of the VRPSD instances. Therefore, this paper presents portfolios of different evolutionary algorithms to reduce the computational time taken to resolve the VRPSD. Moreover, an innovative concept of the mobility allowance (MA) in landmoves based on the levy's distribution function has been introduced to cope with real situations existing in vehicle routing problems. The proposed portfolio approach has been evaluated for the varying instances of the VRPSD. Four of the existing metaheuristics including Genetic Algorithm (GA), Simulated Annealing (SA), Artificial Immune System (AIS), TABU Search (TS) along with new neighbourhood search, are incorporated in the portfolios. Experiments have been performed on varying dimensions of the VRPSD instances to validate the different properties of the algorithm portfolio. An illustrative example is presented to show that the set of metaheuristics allocated to certain number of processors (i.e. algorithm portfolio) performed better than their individual metaheuristics. © 2012 Elsevier B.V.

Keywords

era2015

Disciplines

Engineering | Science and Technology Studies

Publication Details

Shukla, N., Choudhary, A. K., Prakash, P., Fernandes, K. J. & Tiwari, M. K. (2013). Algorithm portfolios for logistics optimization considering stochastic demands and mobility allowance. *International Journal of Production Economics*, 141 (1), 146-166.

Algorithm Portfolios for Logistics Optimization Considering Stochastic Demands and Mobility allowance

N. Shukla¹

¹ Warwick Manufacturing Group,
University of Warwick, Coventry,
CV4 7AL, United Kingdom

A.K.Choudhary²

² Logistics and Supply Chain Research Centre
Management School,
The University of Sheffield, UK

P. Prakash³

³ Irish Centre for Manufacturing Research,
NUI, Maynooth,
Co Kildare, Ireland

K. J. Fernandes⁴

⁴The York Management School,
University of York, UK

M K Tiwari⁵

⁵Department of Industrial Engineering and Management
Indian Institute of Technology Kharagpur,
Kharagpur, India

Algorithm Portfolios for Logistics Optimization Considering Stochastic Demands and Mobility allowance

Abstract

The Vehicle routing problem with stochastic demands (VRPSD) is a well known NP-hard problem. The uncharacteristic behaviour associated with the problem enhances the computational efforts required to obtain a feasible and near-optimal solution. This paper proposes an algorithm portfolio methodology based on evolutionary algorithms, which takes into account the stochastic nature of customer demand to solve this computationally complex problem. These problems are well known to have computationally complex objective functions, which make their solutions hard to find, particularly when problem instances of large dimensions are considered. Of particular importance in such situations is the timeliness of the solution. For example, Apple was forced to delay their shipments of iPads internationally due to unprecedented demand and issues with their delivery systems in Samsung Electronics and Seiko Epson. Such examples illustrate the importance of stochastic customer demands and the timing of delivery. Moreover, most of the evolutionary algorithms, known for providing computationally efficient solutions, are unable to always provide optimal or near optimal solutions to all the VRPSD problem instances within allocated time interval. This is due to the characteristic variations in the computational time taken by evolutionary algorithms for same or varying size of the VRPSD instances. Therefore, this paper presents portfolios of different evolutionary algorithms to reduce the computational time taken to resolve the VRPSD. Moreover, an innovative concept of the mobility allowance (MA) in landmoves based on the levy's distribution function has been introduced to cope with real situations existing in vehicle routing problems. The proposed portfolio approach has been evaluated for the varying instances of the VRPSD. Four of the existing metaheuristics including Genetic Algorithm (GA), Simulated Annealing (SA), Artificial Immune System (AIS), Tabu Search (TS) along with new neighbourhood search, are incorporated in the portfolios. Experiments have been performed on varying dimensions of the VRPSD instances to validate the different properties of the algorithm portfolio. An illustrative example is presented to show that the set of metaheuristics allocated to certain number of processors (*i.e.*, algorithm portfolio) performed better than their individual metaheuristics.

Keywords: Algorithm Portfolio, Metaheuristic search, Vehicle Routing Problem, Stochastic Demand. Genetic Algorithm, Simulated Annealing, Tabu Search

1. Introduction

There are several NP hard problems in the areas of graph theory, scheduling and coding theory for which a computationally efficient solution has not been found or shown to be non-existent (Stinson, 1987). This uncharacteristic behaviour is due to the stochastic nature of problem, its size and complexity. Further, the nonlinear characteristics of the objective function of NP hard problems also contribute to an increase in computational complexity. Common examples of non-deterministic engineering problems with uncertainties include vehicle routing problems with stochastic demands (VRPSD) (Moghaddama *et al.* (2012), Goodson *et al.* (2011), Yang *et al.* (2000)), inventory routing problems (Shukla *et al.* (2012), Agghezzaf *et al.* (2006)), travelling salesman problems (TSP) of varying size (De Berg (2005)), and lot-sizing problems with stochastic demands (Raa (2005)). This paper focuses on developing a decision support methodology for resolving varying instances of the VRPSD problem in a computationally efficient way. Next, we briefly discuss the VRPSDs followed by the need for algorithm portfolios and relevant literature.

The Vehicle Routing Problem (VRP) deals with the transportation of goods and services between geographically dispersed cities or customer locations by means of a fleet of vehicles. Solution to such a problem means determining the best set of possible vehicle routes, servicing all customers and optimising related constraints such as vehicles capacity, time windows, driver's maximum working time etc. VRPs are of major focus in supply-chain systems today, and they are becoming increasingly complex and challenging. For this reason, it has attracted various researchers to develop routing models that are more dynamic, stochastic, and incorporate all constraints, thereby enhancing the computational complexity associated with the objective function.

Although several VRPSD models exist, they mainly consider the routing of the vehicle to be a flat path that ensures smooth movement of the vehicles. Based on the general performance and design of vehicles, approximate velocity are assigned to these vehicles. The assumption of smooth flow of vehicles may be useful in some but not necessarily all cases. Usually, ground movement of vehicles are hampered by areas of uneven terrain, which diminishes the effective movement of the vehicle (or the distance traversed is increased). Therefore, the distance traveled by vehicles must be approximated in close range with a view to use some distribution function that is able to fully map the variations originated due to uneven paths.. This paper models such variations using the concept of mobility allowance that views the land disruptions as an extra distance that the vehicles have to cover. In this paper, levy's distribution function has been used to model various types of land disruptions. The theoretical base of this approach bridges the existing research gap and brings closer a practical solution to VRPSDs by considering and modelling issues of mobility allowance.

It is widely recognized that the VRP is one of the most challenging problems to solve (Lim and Wang (2005)). Conceptually, VRP is an NP hard problem that can be viewed as the combination of the Travelling Salesman Problem (TSP) and the Bin Packing Problem (BPP). It is believed that one may never find a computational technique that will guarantee optimal solutions to larger instances for such problems. Even for small fleet sizes and a moderate number of transportation requests, the planning task is highly complex. The optimal strategies do find their application in resolving academic problems of insignificant dimensions, but the real world problem demands more robust heuristic and metaheuristic approaches to solve such problems in the required time frame. The increasing use of the metaheuristics has dramatically reduced the time taken to resolve these problems without much depreciation in the solution quality.

The main contributions of this study are:

- This research introduces the concept of mobility allowance in vehicle routes to mathematically formulate the terrain uncertainties using the Levy distribution function. Thus providing more logical and mathematical grounds for encountering discrepancies in land moves. A mathematical model for estimating the extra distance traversed by vehicles has been also presented to incorporate the complexities.
- This research provides the mechanism for employing key metaheuristics running on multiple processors for providing effective and efficient solutions than the individually running algorithms.
- The proposed method is a more reliable and efficient decision support methodology in terms of producing quick and reliable solutions for complex and dynamically changing instances of VRPSDs. It takes the advantage of combining metaheuristic search approaches with several processors to arrive at the best solution in minimum computational time. Therefore, this research extends the existing literature in vehicle routing by providing a strong and useful alternative to traditional VRP solutions.

Further, this paper also provides mechanisms for algorithm portfolio design, mathematical & statistical evaluation, and analysis for the vehicle routing problems with stochastic demands and mobility allowance. The rest of the paper is organized as follows: Section 2 discusses literature review and identifies current research gaps. Section 3 presents mathematical modeling of Stochastic Vehicle Routing Problem (VRPSD). Section 4 gives a detailed modeling and approximation on the mobility allowance for the vehicle routings. Section 5 describes the concept of algorithm portfolios, details the implementation of the four basic metaheuristics and their advanced variants with the new neighborhood generators, and explains the experimental background used in this paper. Section 6 details the wider insights of the study to the

comparative performance of the metaheuristics, and analyzes the functioning of the portfolios along with suggestions to optimal solutions. Finally, Section 7 concludes the paper with some discussion on future research directions.

2. Literature review

VRP models mainly focusing on the stochastic nature are described in literature as Stochastic Vehicle Routing Problems (SVRPs) (Stewart and Golden (1983), Secomandi (2000), Bent and Hentenryck (2004), Kenyan and Morton (2003), Mendoza *et al.* (2010), Sungur *et al.* (2008), and Erera *et al.* (2010)). As defined by Stewart and Golden (1983), a VRP is stochastic when the demands at individual customer locations behave as random variables, and the routes must be determined before the values of these random variables become known. Earlier, Clarke and Wright (1964) discussed determination of optimum or near-optimum routing of a fleet of trucks of varying capacities. A neuro-dynamic programming based optimisation method to resolve vehicle routing problem with uncertain customer demands and single vehicle is introduced by Secomandi (2000). Bent and Hentenryck (2004) considered dynamic VRP with time windows and stochastic customers for maximizing the number of customers serviced. Bianchi *et al.* (2005) have introduced hybrid metaheuristics to resolve vehicle routing problems with stochastic demands. Recently, a few topics dealing with VRPSDs were studied such as multi-compartment VRPSDs (Mendoza *et al.* (2010) and Sungur *et al.* (2008)), capacitated VRP with demand uncertainty (Sungur *et al.* (2008)), and VRPSDs with duration constraints (Erera *et al.* (2010)). Goodson *et al.* (2011) examined neighborhood structures for heuristic search based on simulated annealing for the VRPSDs. Moghaddama, *et al.* (2012) proposed advanced Particle Swarm Optimization (PSO) algorithm for solving uncertain VRP in which the customer's demands are supposed to be uncertain with unknown distributions. In essence, stochastic nature of VRPSDs are due to a variation in the set of customers visited, demand levels, travel time, etc. These stochastic variables have known probability distributions and the objective function is to minimise the expected cost of the planned routes. VRPSDs have been studied under various formulations and a survey of these models is provided in Gendreau *et al.* (1996).

Many random search algorithms such as Genetic Algorithm (Gen and Cheng (2003), Baker and Ayechev (2003)), Tabu Search (TS) (Glover *et al.* (2005) and Simulated Annealing (Kirckpatrick (1983)) have been extensively used and their robustness and applications are well established in the VRP literature. Several metaheuristic approaches, mixed with efficient local and neighborhood search techniques have also been proposed to resolve large scale VRPs (Bianchia *et al.* (2005)). These approaches are introduced in the literature to minimize the response time of the method adopted and the quality of solution found. These metaheuristics have been extensively used for solving the VRPSD problem and problems of similar

complexity.

Park (2001) proposed a hybrid genetic algorithm (HGAV) incorporating a greedy interchange local optimization algorithm for the vehicle scheduling problem with service due times and time deadlines, three conflicting objectives of the minimization of total vehicle travel time, total weighted tardiness, and fleet size are considered. The results show that the HGAV always attains better solutions than the BC-saving algorithm, but with a much longer computation time (Hong and Park (1999)). Wasner and Zapfel (2004) described an optimal design of depot and hub transportation networks for parcel service providers. A generalized hub location and vehicle routing model is developed which encompasses the determination of the number, locations of hubs and depots, their assigned service areas, and the routes between demand points and consolidation points (depots, hubs). Hong and Park (1999) formulated a bi-objective vehicle routing problem with time window constraints to determine the most favorable vehicle routes that minimize the total vehicle travel time and the total customer wait time. They constructed a linear goal programming (GP) model for the problem, and proposed a heuristic algorithm to relieve a computational burden inherent to the application of the GP model. Bernard *et al.* (2008) developed and analyzed a mathematical model for dynamic fleet management that captures the characteristics of modern previous term vehicle next term operations. Herer *et al.* (1997) developed an approach for Metered Inventory Routing Problem to minimize the total discounted costs using the concept of ‘temporal distances’ and used a simulation study to demonstrate the effectiveness of the procedure. Hwang (2005) discussed an integrated distribution previous term routing next term problem for multi-supply centers based on improved genetic algorithm and graphical user interface (GUI)-type programming. Zapfel and Bogl (2008) presented a real-life case study that considers short-range weekly planning of postal companies that must decide about pickup and delivery tours for fluctuating volume (number of shipments), time windows for demand points, variable vehicle capacities, personnel planning, and outsourcing decisions for tours and drivers. They proposed a hybrid metaheuristic combined with a construction heuristic and demonstrated it using a computer simulations have demonstrated. Jin *et al.* (2007) proposed a two-stage algorithm with valid inequalities to optimally solve the split delivery vehicle routing problem. The numerical experiments showed that their approach significantly outperforms other exact solution approaches provided in the literature for this problem. Li *et al.* (2010) studied a variant of stochastic vehicle routing problems, in which travel and service times are stochastic, and a time window constraint is associated with each customer. They formulated it as a chance constrained programming model and a stochastic programming model, and applied a Tabu search method to solve them. Kritikos and Ioannou (2010) discussed a variant of the vehicle routing problem with time window that targets the balancing of the load carried by each active vehicle. They proposed a method based on free disposal hull and data envelopment analysis. They claim to produce very good results for the

mentioned problem based on computational experimentation.

Most of the aforementioned approaches have used metaheuristics to solve the underlying problem. However, the performance of a metaheuristic heavily depends upon random response samples generated during the course of their run, which causes large variations in the quality of the solution obtained. As a result, metaheuristics when used individually may provide optimal or near optimal solutions for some instances of a particular problem. However, use of individual metaheuristics cannot be considered as an ‘optimal’ approach for the entire range of instances of VRPSD. Therefore, this paper details a novel concept of algorithm portfolios, which can be used to resolve computationally hard VRPSD problems. This paper presents generic applicability and statistical background to validate the concept of algorithm portfolios based on metaheuristics.

A algorithm portfolio can be defined as "*a collection of different algorithms and/or different copies of the same algorithm running on different processors*" (Brown *et al.* (2003)). The need for constructing the algorithm portfolio is due to variations in performance of several metaheuristics for the same problem of varying complexity or dimensions, or using different random seeds on the same problem. One can benefit by such variations, by combining several algorithms to form algorithm portfolios, and running them in parallel or interleaving them on a single processor. Shukla, *et al.* (2012) present algorithm portfolios based on variants of Genetic Algorithm (GA) to deal with the inventory routing problem of supply chain. This paper introduces an algorithm portfolio concept based on metaheuristics, to solve complex problems such as VRPSD. The proposed algorithm portfolios are constructed using eight metaheuristics that consist of four basic metaheuristics that include: GA, SA, TS, AIS, and their enhanced variants *equipped with new neighborhood generation schemes* (represented as EGA, ESA, ETS, EAIS), ensuring a more robust experimental ground. A comprehensive and rigorous experimentation is conducted in this paper to evaluate the performance of various algorithm portfolios. The best algorithm portfolio cases corresponding to a set of processors are determined using Analytical Hierarchy Process (Winkler (1990)). Moreover, selection of the best performing algorithm within the algorithm portfolio for a given optimization problem is obtained using the hamming distance based algorithm selection procedure. In addition, a comparative analysis of the best algorithm portfolio cases and the best performing constituent algorithm has been made over ten VRPSD problem instances with varying size and complexity. Next section discusses mathematical modeling of the VRPSD problem and related constraints.

3. Mathematical Modeling of VRPSD

A VRPSD is defined on a complete graph $G = (V, A, C)$, where $V = \{0, 1, \dots, n\}$ is a set of nodes *i.e.*, (customers) with node 0 denoting the depot, $A = \{(i, j) : i, j \in V, i \neq j\}$ is the set of arcs joining the nodes, and $C = \{c_{ij} : i, j \in V, i \neq j\}$ are the travel costs (distances) between nodes. The cost matrix C is symmetric and satisfies the triangular inequality. A fleet of TNV homogeneous vehicles, each with capacity Λ has to deliver goods to n customers according to their stochastic demands, minimizing the total expected distance traveled, and given that the following assumptions are made.

Customer's demands are stochastic variables $\theta_i, i=1, 2, \dots, n$ independently distributed with known distributions based on past demands. The actual demand of each customer is only known when the vehicle arrives at the customer location. It is also assumed that θ_i does not exceed the vehicle's capacity Λ , and follows a discrete probability distribution $\pi_{id} = \text{Prob}(\theta_i = d), d = 0, 1, 2, \dots, D \leq \Lambda$ where π_{id} is the probability of i^{th} stochastic variable associated with d demand. A solution to the VRPSD is a permutation of the customers and dummy depots (represented by 0 and equal to the number of vehicles) that forms a cyclic TSP string of the underlying problem (see Fig. 1). All the vehicles have to visit the customers in the order specified by the priori tour. The k^{th} vehicle has to traverse n_k customers in the order defined by priori tour. According to the customers actual demand, the vehicle decides whether to proceed to the next customer or to go to depot for restocking. Sometimes the choice of restocking is the better option, even if the vehicle is not empty, or if its capacity is bigger than the expected demand of the next scheduled customer, this action is termed as *preventive restocking*. The goal of preventive restocking is the backtracking of the vehicle when they do not have sufficient load to serve the preceding customer, thus, it performs a back and forth trip to the depot for completing the delivery. Let us define some of the variables that are crucial in formulating the constraints:

$X_{i,j,k}$: binary flow variable, equals 1 if path (i, j) is traversed by vehicle k and 0 otherwise.

$Y_{i,j,k}$: amount of commodity that is traversing arc (i, j) by the means of vehicle k .

s_{ik} : amount of commodity supplied at the i^{th} customer by vehicle k

M : a large constant to transform nonlinear terms to linear ones for some constraints

TNV : total number of vehicles initially available at the depot.

Insert Figure 1 about here

The objective function, *i.e.* the expected distance covered by the vehicle, is computed as follows. Let \mathfrak{R} represent the priori tour (defined earlier) traversed by the TNV vehicles, and the expected cost of the priori

tour traversed by k^{th} vehicle is $f_{0k}(\Lambda)$. Therefore, the total expected cost for traversing the priori tour by all the TNV vehicles is denoted by OBJ . Now, if L_{jk} represents the set of all possible loads that the k^{th} vehicle can have after the completion of the service at the j^{th} customer, then, $f_{jk}(\lambda) \forall \lambda \in L_{jk}$ satisfies,

$$f_{jk}(\lambda) = \text{Minimum} (f_{jk}^p(\lambda), f_{jk}^r(\lambda)) \quad (1)$$

where,

$$f_{jk}^p(\lambda) = c_{j,j+1} + \sum_{d:d \leq \lambda} f_{j+1,k}(\lambda - d)\pi_{j+1,d} + \sum_{d:d > \lambda} [2c_{j+1,0} + f_{j+1,k}(\lambda + \Lambda - d)\pi_{j+1,d}] \quad (2)$$

$$f_{jk}^r(\lambda) = c_{j,0} + c_{0,j+1} + \sum_{d=1}^D f_{j+1,k}(\Lambda - d)\pi_{j+1,d} \quad (3)$$

The pseudo code for calculating the objective function is presented in Figure 2.

Insert Figure 2 about here

with the boundary condition $f_{n_k}(\lambda) = c_{n_k,0}, \lambda \in L_{n_k}$. In the above Eqs. (2-3), $f_{jk}^p(\lambda)$ is the expected cost corresponding to the choice of proceeding directly to the next customer. Whereas, $f_{jk}^r(\lambda)$ is the expected cost of preventive restocking by the k^{th} vehicle. As detailed by Yang *et al.* (2000), given a priori tour, for each j^{th} customer, there is a load threshold LT_{jk} for vehicle k such that, if the residual load after serving the j^{th} customer is greater than or equal to LT_{jk} , then it is better to proceed to the next planned customer. Otherwise, it is better to go back to the depot for preventive restocking.

$$\text{Minimize } \{ OBJ \} \quad (4)$$

Where, $OBJ = \sum_{k=1}^{TNV} f_{0k}(\Lambda)$ as illustrated in Fig. 2. Subject to the following constraints:

$$\sum_{k=0}^{TNV} X_{0,i,k} \leq TNV \quad \forall i \in V \quad (5)$$

$$\sum_{j=1}^{|V|} X_{0,j,k} = 1 \quad \forall k \in \{0, 1, 2, \dots, TNV\} \quad (6)$$

$$\sum_{j=1}^{|V|} X_{j,0,k} = 1 \quad \forall k \in \{0, 1, 2, \dots, TNV\} \quad (7)$$

Constraint Eqs. (5-10) characterizes the vehicle flow on the path. Equation 5, illustrates that the number of vehicles to service must not exceed the available vehicles ready at the depot. Equations (6-7) represents the constraint of each vehicle flow from and back only to the supply node (denoted by 0).

$$\sum_{j=0}^{|V|} \sum_k^{TNV} X_{i,j,k} = 1 \quad \forall i \in V \quad \text{and } i \neq 0 \quad (8)$$

$$\sum_{j=0}^{|V|} \sum_k^{TNV} X_{j,i,k} = 1 \quad \forall i \in V \quad \text{and } i \neq 0 \quad (9)$$

$$\sum_{j=0}^{|V|} X_{i,j,k} = \sum_{j=0}^{|V|} X_{j,i,k} \quad \forall i \in V \quad k \in \{0,1,2,\dots,TNV\} \quad (10)$$

$$s_{ik} - \left[\sum_{j=0}^{|V|} Y_{i,j,k} - \sum_{j=0}^{|V|} Y_{j,i,k} \right] \geq 0 \quad \forall i \in V \quad k \in \{0,1,2,\dots,TNV\} \quad (11)$$

$$X_{i,j,k} \Lambda \geq Y_{i,j,k} \quad \forall i, j \subseteq V \quad k \in \{0,1,2,\dots,TNV\} \quad (12)$$

Equations (8-9) state that each demand node must be visited exactly once. Equation 10 infers that all vehicles that flow into the customer node must also flow out of it. Equation 11 provides the balanced commodity flow requirement of the depot. Equation 12 allows the flow of the commodities as long as there is sufficient vehicle capacity.

$$\sum_{j=0}^{|V|} X_{i,j,k} \Lambda \leq M \left[\sum_{j=0}^{|V|} Y_{i,j,k} + \sum_{j=0}^{|V|} Y_{j,i,k} \right] \quad \forall i \in V, i \neq 0, k \in \{0,1,2,\dots,TNV\} \quad (13)$$

$$X_{i,j,k} = 0,1 \quad Y_{i,j,k} \geq 0 \quad (14)$$

$$\sum_{k=1}^{TNV} n_k = n \quad (15)$$

Equation 13 states that whenever a vehicle visits a demand node, it should carry some commodities either from or to that node. These aforementioned constraints, establish the connection between the commodity flow and vehicle flow. Equation 14 states the non-negativity properties of the decision variables. Constraint Equation 15 illustrates that sum of the customers traversed by TNV vehicles must be equal to n . The following section describes the mathematical modelling and evaluation of mobility allowance in the vehicle land moves or routes.

4. Mobility Allowance in Land Moves

The mobility allowance is an innovative concept introduced into the vehicle routine problem with stochastic demands. The innovative twist, is that the vehicles are allowed to deviate from the given route. Usually, the land moves and distances taken up in most of the research, even though straight, cannot be

avoided from having small disruptions. Thus, the interpretation of the routes have to be modified to map the real situations. These variations are attributed to various factors like uneven terrain, natural hindrances, etc. In this research, issues of disruption are considered by adding mobility allowance to each vehicle route. To achieve the aforementioned objective, a formal levy distribution function that includes a wide range of modeling flexibility has been utilized. The following subsection, presents the details of the levy distribution function on which the allowance analysis has been done.

4.1 Levy's Probability Distribution

Generally, unexpected variations are approximated by a normal distribution with a given mean and standard deviation as per the abnormality of the model. Out of three distribution functions (Gaussian, Cauchy and Levy), *Gaussian* distributions are characterized by finite second moments and are stable. Unlike, gaussian probability distribution, the Cauchy probability distribution has infinite second moments and, as a result, has much longer tails. However, Levy's Probability Distribution has infinite second moments and are yet stable. An infinite second moment implies the absence of a definite scale. In addition, the Cauchy distribution is a special case of Levy's distribution (Lee and Yao (2003)). The following equation defines Levy's Probability Distribution:

$$F_{levy}^{\alpha\gamma}(y) = \frac{1}{\pi} \int_0^{\infty} e^{-\gamma q^{\alpha}} \cos(qy) dq \quad y \in R \quad (16)$$

It can be seen after plotting the abovementioned distribution function, that the distribution is symmetric with respect to $y = 0$. This distribution has two control factors α ($0 < \alpha < 2$) responsible for the shape, with $\gamma > 0$ as a scaling factor. Most of the stable distributions can be approximated as special cases for the above function. The distribution resembles a Cauchy distribution when $\alpha = 1$ and the approximate Gaussian distribution with $\alpha \rightarrow 2$. Various shapes and properties of the function can be obtained by varying these two parameters depending upon the requirements. In this manner, α controls the shape of distribution in such a manner that the different shapes of probability distributions can be achieved in the tails, the smaller the value of α the longer the tail. Here, it can be seen that a Levy distribution function comprises of a large number of variations depending upon the parameters. Thus, it will be shown later, a distribution can be approximated from Eqn. (16) that may map the path deviation considering the extent of deviations.

Lemma 1 *For a Levy Distribution, if the value of a γ is known, it is possible to obtain the distribution for any other γ , thus, making it a scaling factor.*

Proof 1: Let y be scaled to cy (where c is a constant). Replacing it in equation (16), without loss of generality, the new scaled equation can be stated as:

$$F_{levy}^{\alpha\gamma}(cy) = \frac{1}{c\pi} \int_0^\infty e^{-\gamma'q^\alpha} \cos(qy) dq = \frac{1}{c} F_{levy}^{\alpha\gamma'}(y) \quad (17)$$

Thus, any γ' can be represented in terms of γ as $\gamma' = \gamma c^{-\alpha}$. Replacing the value of c in equation 17 with $\gamma' = 1$, the following relationship is obtained:

$$F_{levy}^{\alpha\gamma}(y') = \gamma^{-\frac{1}{\alpha}} F_{levy}^{\alpha,1}(y). \quad (18)$$

Since the distribution for any other γ can be obtained from the distribution $\gamma = 1$, it can invariably be treated as a scaling factor. This property of *Levy's Distribution* is very helpful in calculating the expected value of change in the length of a path, while introducing the mobility allowance to the vehicle. The details are discussed in next subsection.

4.2 Mobility Allowance for Vehicle Routing

This section details the increment in length of the vehicle tour that arise from uncertain variations of the vehicle when travelling. Let the vehicle moving between any two points (or customer locations) and the distance between these points be L , calculated as the Euclidean distance between the two coordinate locations. The route length is subject to variations depending upon various factors, thus, causing an increase in the effective length of the tour and increasing the time (t) taken to cover it. If extra deviation occurs due to disruptions in the route Δ , then index ε can be defined as:

$$\varepsilon = \frac{L}{L + \Delta} \quad (19)$$

This fraction ε is a characteristic associated with a path, not with any particular vehicle and is to be added to the length of the route. In fact, since the velocity has changed, it is indispensable to change the mileage of the vehicle in order to realize the fuel consumption due to increased length.

Let the mileage of the vehicle over the path L be mg , and ε' be the corresponding decrease in mileage. The following relationship exists:

$$\begin{aligned} \text{Total fuel consumption} &= \frac{L}{mg} = \frac{L + \Delta}{\varepsilon' \times mg} \\ \Rightarrow \quad \varepsilon' &= \frac{L + \Delta}{L} \quad \text{i.e.,} \quad \varepsilon' = 1 / \varepsilon \end{aligned} \quad (20)$$

Therefore, again no extra computational burden is introduced. Now, let an expected value of change in length of a path based on Levy's distribution be calculated. First, without the loss of generality, it is assumed

that the majority of path deviations are confined to a rectangle with length L and width w . The width w , can be obtained as an input based on general experiences and past knowledge about the tours. In this research, it is assumed that $\wp = 99\%$ of the deviations in the path are confined within width w . The factor \wp is a tuning parameter. 1% probability is left to simulate the situations that are encountered unexpectedly giving rise to undesired lengths. The plausible movement of a vehicle within the conceptualized rectangular region can be shown in Fig. 3.

Insert Figure 3 about here

Here, the straight line connecting the two points is taken as the mean, and the Levy distribution is used to generate the random points within the rectangle shown. Next, the probability of generating a number utilizing Levy distribution in the range $[-\frac{w}{2}, \frac{w}{2}]$ is explored and is equated to \wp .

Lemma 2: *The probability to generate a random number within the range $[-\frac{w}{2}, \frac{w}{2}]$ is given by:*

$$P_{levy}(|y| \leq \frac{w}{2}) = \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^n \frac{n(\alpha+1)+1}{\alpha} \frac{(2n+1)}{\alpha \times (\alpha)!} \times \left[\frac{w^{2n+1}}{2^{2n}(2n+1)} \right] \quad (21)$$

Proof 2:

$$P_{levy}(|y| \leq \frac{w}{2}) = \int_{-\frac{w}{2}}^{\frac{w}{2}} F_{levy}^{\alpha, \gamma}(y) dy \quad (22)$$

Therefore, the task is to compute $F_{levy}^{\alpha, \gamma}(y)$. Taylor expanding cosine in Eqn. (17)

$$F_{levy}^{\alpha, \gamma}(y) = \frac{1}{\pi} \int_0^{\infty} \sum_{n=0}^{\infty} (-1)^n \frac{(qy)^{2n}}{(2n)!} e^{-\gamma q^{\alpha}} dq \quad (23)$$

substituting $\kappa = \gamma q^{\alpha}$, and $d\kappa = \alpha \gamma q^{\alpha-1} dq$. For the notational simplicity, the symbol Γ is used to denote forthwith $F_{levy}^{\alpha, \gamma}(y)$.

$$\begin{aligned} \Gamma &= \frac{1}{\pi} \sum_{n=0}^{\infty} \int_0^{\infty} (-1)^n \frac{(y)^{2n}}{(2n)!} \left[\frac{\kappa}{\gamma} \right]^{2n/\alpha} e^{-\kappa} \frac{d\kappa}{\alpha \gamma \left[\frac{\kappa}{\gamma} \right]^{\frac{\alpha-1}{\alpha}}} \\ &= \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^n \frac{(y)^{2n}}{\alpha (2n)! \gamma^{\frac{\alpha-1}{\alpha}}} \int_0^{\infty} \kappa^{\frac{(2n+1)}{\alpha}} e^{-\kappa} d\kappa \end{aligned} \quad (24)$$

Integrating by parts:

$$\begin{aligned}
\Gamma &= \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^n \frac{(y)^{2n}}{\alpha(2n)! \gamma^{\left(\frac{2n+1}{\alpha}\right)}} \left[\kappa^{\frac{2n+1}{\alpha}} \int_0^{\infty} e^{-\kappa} d\kappa \right] - \frac{2n+1}{\alpha} \int_0^{\infty} e^{-\kappa} \cdot \kappa^{\frac{2n+1}{\alpha}-1} d\kappa \\
&= \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^n \frac{(y)^{2n}}{\alpha(2n)! \gamma^{\left(\frac{2n+1}{\alpha}\right)}} \times (-1)^{\frac{2n+1}{\alpha}} \left(\frac{2n+1}{\alpha} \right)! \\
&= \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^{\frac{n(\alpha+1)+1}{\alpha}} \frac{(2n+1)(y)^{2n}}{\alpha \times (\alpha)! \gamma^{\frac{2n+1}{\alpha}}} \tag{25}
\end{aligned}$$

Without loss of generality (from Lemma 1), the value of γ can be taken as 1. Considering probability equal to 0.99 and substituting the values in Eqns. (22) & (25), the following equation can be derived:

$$\begin{aligned}
\wp &= \int_{-\frac{w}{2}}^{\frac{w}{2}} \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^{\frac{n(\alpha+1)+1}{\alpha}} \frac{(2n+1)(y)^{2n}}{\alpha \times (\alpha)!} dy \\
\wp &= \frac{1}{\pi} \sum_{n=0}^{\infty} (-1)^{\frac{n(\alpha+1)+1}{\alpha}} \frac{(2n+1)}{\alpha \times (\alpha)!} \left[\frac{w^{2n+1}}{2^{2n}(2n+1)} \right] \tag{26}
\end{aligned}$$

This equation can be used to approximate the value of α that would result in a probability distribution having 99% of the expected values generated within the range $[-\frac{w}{2}, \frac{w}{2}]$. Such an approximation can be done using powerful approximation packages readily available, however, the actual approximation considered in this research is given in section 6.1.

The basic motive of the above formulation and derivation, was to approximate the increase in length for a path of length L . To continue the theme, let us consider the fact that the complete path is divided into various smaller rectangles as shown in Fig. 3, each having width w and length ℓ . Therefore, the number of such rectangles is L/ℓ . In this manner it can be seen from Eqs. (17) and (26) that the distance between the points to be covered in the path is calculated for a rectangle. Now we extend this result to complete length L .

Lemma 3: *Given a large number I , the expected length of a route in the segment with length ' ℓ ' can be given as:*

$$E_{levy}\{\Xi\} = \int_{-I}^I \left[2\sqrt{y^2 + \left(\frac{L}{\ell}\right)^2} \frac{1}{\pi} \int_0^{\infty} e^{-\eta q^{\alpha}} \cos(qy) dq \right] \tag{27}$$

Proof 3: Let us consider that the diagonal \hbar as shown in Fig. 3, represents the movement of a vehicle within the rectangle with length ℓ , and width as a mean step size. The approach is to calculate the expected distance traversed as the sum of the mean distance covered for ℓ parts between the end points given in

Fig. 3. The expected distance moved for any segment of the path can be calculated as:

$$\text{Expected length } \Xi \text{ for segment } \xi = 2 \times \sqrt{y^2 + \left(\frac{L}{\ell}\right)^2} \quad (28)$$

The expected length that vary according to the aforementioned Levy Probability distribution function (Eq. 17), can be calculated as:

$$E_{levy}\{\Xi\} = \int_{-\infty}^{\infty} \left[2 \times \sqrt{y^2 + \left(\frac{L}{\ell}\right)^2} \right] \frac{1}{\pi} \int_0^{\infty} e^{-|q|^\alpha} \cos(qy) dq \quad (29)$$

Since, the above integral is not finite, without loss of generality and for practical purposes, the approximation can be made for user defined larger value I (calculated later in the experimental sections). In this manner, a more practical assumption to expected length would be:

$$E_{levy}\{\Xi\} = \int_{-I}^I \left[2 \times \sqrt{y^2 + \left(\frac{L}{\ell}\right)^2} \right] \frac{1}{\pi} \int_0^{\infty} e^{-|q|^\alpha} \cos(qy) dq \quad (30)$$

Now, combining Eq. (20) with Eq. (30), the total distance ($L + \Delta L$) covered by a vehicle between the two points separated by Euclidean distance L , can be calculated as:

$$L + \Delta L = E_{levy}\{\Xi\} \times \frac{L}{\ell} \quad (31)$$

Thus, the fraction increase in the Euclidean distance \mathcal{E} is now defined as:

$$\mathcal{E} = \frac{\frac{L}{\ell} \times E_{levy}\{\Xi\}}{L} = \frac{E_{levy}\{\Xi\}}{\ell} \quad (32)$$

This approximated increase in Euclidean distance, influences the traveling cost c_{ij} of the vehicle traveling from the i^{th} node to j^{th} node. To retain parsimony, similar approximations on land moves has been done for all arcs (i, j) . As a theoretical framework, this research proposes to count such disruptions based on the Levy distribution. Such base is deemed to be more efficient, and is expected to provide a more robust foundation to deal with the complexities of the underlying vehicle routing problem. Considering these complexities, the calculation is proposed to be offline and can be further extended to the online uncertainties. The next section describes the context and utilization of the Algorithm portfolio concept, to solve underlying stochastic vehicle routing problem with mobility allowance routes.

5. Portfolio Design

5.1 Problem Characteristics and Instances Explored

In the absence of any commonly used benchmarks for VRPSD problems we have generated our own test problems. These problem instances are generated by controlling four factors that govern the difficulty of the VRPPSD and include: (a) customer locations, (b) capacity over demand ratio, (c) variance of stochastic

demand, and (d) number of customers. The position of customers are randomly assigned in a uniformly discretized space.

In this research, three different problems are designed by varying four factors with a view to achieve two different goals: i) to analyze the metaheuristics performance under different stochastic environments, and ii) to study the performance of enhanced versions of the metaheuristics. In this research, we consider three different instances (small, medium, and large) with 50, 100 and 200 distributed customers. In the instances, the position of customers is chosen at random in the uniform discretized space of $[0,100]^2$. The most important of all the factors is total (average) demand to vehicle's capacity ratio that influences the difficulty of the problem instances (Gendreau, 1995). Customer demand is a stochastic variable, uniformly distributed over an interval of predefined size. Two parameters which determine the stochastic demand of the j^{th} customer include: average demand (D_j); and the spread (S_j), so that the demand for the j^{th} customer is $[D_j - S_j, D_j + S_j]$. In all the three instances, the average demand for all customers varies between $[0,49]$ or $[50,100]$, and the spread is considered to be in the range of $[5,15]$. The average number of customers served before restocking is fixed to 4, and therefore total demand over vehicle capacity ratio remains within the range of $[12, 50]$. The number of vehicles in all the three cases (small, medium, and large) are 8, 16, and 24. The solution to these problems has been attempted by using an algorithm portfolio that works on the strategy of minimizing the risk in terms of computational cost and the solution quality obtained. The following subsection details the design of proposed portfolios.

5.2 The Portfolio Design

Although, some of the algorithms usually perform better than others, it is not possible to define a best algorithm for a given problem set which will work on all instances of the problem. Different problem instances are usually associated with varying computational time (Brown *et al.* 2003). It is well known that the *NP*-Hard problems furnish much pronounced variations in run times and the solution quality of algorithms. Thus, amidst such variations, choosing the best algorithm for a particular problem is a hard decision. Initial research to provide a tentative solution to such a dilemma goes back to seventies when Rice (1976) characterized it as an algorithm selection problem. Ironically, the research focus failed to concentrate over the algorithm selection perspective, only the last few years have witnessed a re-focus on the problem; thereby producing some excellent, though not comprehensive literature for researchers and practitioners (Gomes and Selman (2001)).

Mostly, the algorithm selection was attempted using a "winner take all" strategy, *i.e.* select a best performing algorithm based on its performance over the given problem in various attempts (Rice (1976)). The major drawbacks of this strategy are highlighted when algorithms are considered which produce better

than average performance on a given instance, whereas providing no better solutions in all other instances. Such scenarios can be often seen in Vehicle Routing problems where the decision maker has to work on various scenarios. They also have to face problems of varying dimensionality. Thus, the problem of choosing a best alternative solution to get the task done is always a crucial decision making problem. This motivates the authors to conceptualize the design of algorithm portfolios in order to get a viable and insightful perspective to the solution of the vehicle routing problem with stochastic demand and routing mobility allowances.

The algorithm portfolios can be considered as a complete bundle of algorithms tied together and are executed on different processors. The urge to improve the performance of algorithms in terms of expected computational cost and solution quality motivated the use of algorithms in the algorithm portfolio. The following sub-section details various aspects of portfolio design and performance measures to assess the working of various portfolios.

Portfolio Design, basically targets to minimize the expected risk and maximize solution advantage in terms of computational cost and deviation from expected. The term *efficient frontier*, is defined for the portfolios performing competitively on the basis of performance measures described. The efficient frontier is always helpful to enhance the decision flexibility of the system and above all maintaining a repository of elite portfolios for future investigations.

In this research, portfolios consist of eight algorithms, or more specifically four metaheuristics and their counterparts equipped with the proposed neighborhood generators. Early attempts to solve the VRPSD were concerned in using a single evolutionary algorithm (Gendreau (1996), Teodorovic and Pavkovic (1992)) resulting in the selection of biased algorithm for training and testing instances. This research presents a novel approach of collecting different algorithms into a portfolio. The logic is to provide the decision makers preferred alternatives, and exploring the applicability of other strategies in solving the class of instances. For example, in a scenario where a decision maker needs a good solution quickly, rather than an optimal solution to avoid computational burden. In addition, the commonly used OROpt solution methodology (Yang *et al.* (2000)) is not always time efficient and more often even initial feasible solutions requires long computation. An obvious choice is to search for other strategies. Thus, intensifying the need for portfolios with all required adaptabilities. The next sub-section briefly describes the algorithms utilized in the portfolio design, followed by a description about performance measures related to portfolio assessment. Finally, the section ends with the experimental design for the proposed portfolio analysis.

5.3 Algorithmic Suit

In order to implement the investigated algorithms, their well known and enhanced variants have been exploited in this research. Three types of neighborhood search algorithms are considered in the portfolio to

escape the local optima and to ensure fast convergence of the portfolio. All the algorithms used in this research require similar representation schema. Therefore, the following discussion presents a brief overview of the implementation of algorithms. Section 5.3.1 details the encoding schema used in the proposed methodology. A detailed study of different algorithms can be found in Section 5.

5.3.1 Encoding Schema

The solution string is generated by considering the number of customers and vehicles allotted. A cyclic Travelling Salesman Problem (TSP) string is formed by considering several dummy nodes of the depot. Each dummy node characterizes the route taken by a particular vehicle. An example of encoding schema used in this research is shown in Fig. 1.

Insert Figure 4 about here

5.3.2 Neighborhood Generation

A fundamental aspect of the the performance of all the search strategies, is to identify an effective neighborhood for defining moves from one solution to another. The search metaheuristic attempts to enhance the quality of the solution by using compound moves. The basic moves consist of a simple insertion or exchange of vertices/arcs on the graph of the problem, compound moves are usually obtained by the combination of basic moves. Osman (1993) used a combination of insertion moves and exchange moves, vertex shifts from one route to another, and exchange vertices between routes based on a 2-opt process. In this research, the following three neighborhood generations schemes (Rego (2000)) have been used:

- **Multiple Exchanging Scheme:** In this type of move, several triplets of the node are taken from the SVRP solution string (formulated as cyclic TSP), then multiple exchanging of the nodes (including customer and dummy nodes) are done using the following procedure:

There are a triplets, selected from the solution string. The k^{th} triplet is defined by $(g_k^{-1}, g_k^0, g_k^{+1})$ that include two arcs (g_k^{-1}, g_k^0) and (g_k^0, g_k^{+1}) . Set T can be defined as:

$$T \equiv \bigcup_{k=0}^{a-1} \{(g_k^{-1}, g_k^0, g_k^{+1})\} \quad (33)$$

representing a triplets. The process of multiple exchange is done by transforming T into T' , where:

$$T' \equiv \bigcup_{k=1}^{a-1} \{(g_k^{-1}, g_{k-1}^0, g_k^{+1})\} \cup \{(g_0^{-1}, g_{a-1}^0, g_0^{+1})\} \quad (34)$$

The transformation of T i.e. $T \rightarrow T'$ shows a replacement of the central node of every triplet. Figure 4 (Left) illustrates the schematic diagram showing the multiple exchanging scheme.

- **Multiple Insertion Scheme:** This scheme consists of choosing a set (g_p^{-1}, g_q^0) in whole graph, excluding the nodes that are already present in set T . Then replacing it by another set in the T . Figure 4 (Middle) illustrates this concept. In this manner, by performing this move, triplet T is transformed into T'' , where T'' is defined as:

$$T'' \equiv \bigcup_{k=1}^{a-1} \{(g_k^{-1}, g_{k-1}^0, g_k^{+1})\} \cup \{(g_p^{-1}, g_{a-1}^0, g_q^{+1})\} \quad (35)$$

- **Multiple Reversal Scheme:** In this neighborhood scheme, part of a solution string is taken out and added again to the solution string by reversing the node order. An illustration of this scheme is presented in the Fig. 4 (Right).

These neighborhoods are generated sequentially as to obtain better solutions for the problem. These neighborhood generation schemes are used by the following algorithms present in the portfolio analysis.

5.3.3 Genetic Algorithm (GA)

Genetic Algorithm incorporated in the portfolio is adopted from (Baker and Ayechev (2003)) with additional features of more explorative neighborhood search strategies. The basics of GA are well known, i.e., a population \mathbf{P} solutions is maintained from which parents are selected for the reproduction processes (crossover and mutation). In the present context, the crossover and mutation operators have been applied to create valid solution strings for the next population.

Offspring O , carry the features of parents. The "survival of fittest" strategy is applied probabilistically for maintaining an individual's fitness. The only difference between the GA adopted by (Baker and Ayechev (2003)) and the one used in this research is the use of neighborhood generation schemes defined in the previous section 5.4.2. They are used for the best solution found in each generation in order to have an explorative search in the preferred regions.

5.3.4 Simulated Annealing (SA)

Simulated Annealing resembles the process of physical annealing of solids (Kirckpatrick (1983)); in computational terms the process can be summarized as starting from an initial solution, exploring its nearby solutions by a perturbation process, and then if obtained, replacing the solution with higher energy solutions. In this process, the less energy solutions are also given a probable chance to be chosen in order to avoid any chance of entrapment in local optima and losing the best solution. Here, a nearby solution slightly differs from the current solution, similar to a small displacement of an atom in material to gain a

new configuration. The relative change in objective function value provide the respective energy change δ . The inferior solutions are assigned a probability of $e^{-\frac{\delta}{(K_b t)}}$, where K_b is the Boltzmann constant and t is the simulated temperature. The use of the Boltzmann function requires the generation of a sufficient number of perturbed solutions to justify its use (Breedam (1995)). Therefore, at each temperature decrease, few iterations m' are run with same temperature. The temperature is decreased according to the relation $t' = \zeta \times t$, where t is current temperature, t' is decreased temperature and ζ is decrement constant.

Although several variants of SAs have been used to solve the VRPs, this research adopts the mechanism proposed by (Breedam (1995)) with enhanced neighborhood generation schemes mentioned in Section 5.4.2 to ensure fast convergence. The neighborhood generation procedure is invoked in a sequential procedure to the best solution found after m' iterations *i.e.*, best solution at a particular temperature, and only the moves that result in better solutions are accepted.

5.3.5 TABU Search

TABU Search used in the portfolio is adopted from Brandao and Mercer (1997). They used TABU search algorithm for solving the multi trip vehicle routing and scheduling problem. The major difference between the TABU search algorithm as proposed by Brandao and Mercer (1997) and this research is the use of search strategy. This research utilizes the neighborhood search strategy as presented in Section 5.4.2 in place of the simple *insert* and *swap* moves as used in the original TABU search algorithm.

The search starts with a feasible solution s , and proceeds through three phases of improvement and selection: (i) phase I: is characterized by exploring the neighborhood (section 5.4.2) for a given number of iterations or until a better solution is not obtained. The best solution (**Best^{TABU}**) is checked with the TABU list containing the elite solutions; and if the move is permissible, the solution is passed on to phase II; (ii) phase II: aims to re-explore the neighborhood for the predefined number of iterations or until a better solution is not obtained. The completion of the phase sets off a more tight bounded; and (ii) phase III: explores infeasible solutions in the feasible ranges utilizing the neighborhood moves. The TABU search is distinguished by an exploration of neighborhood, but are usually bounded by longer run times.

5.3.6 Artificial Immune Systems (AIS)

Artificial Immune System (AIS) is a relatively newer approach to optimization as compared to the aforementioned three strategies. AIS incorporates clonal selection with evolutionary principles as proposed by Castro and Zuben (2002). It can be treated as an established metaheuristic with a number of superior results to its credit (De Castro and Timmis (2002)). AIS is also a population based search strategy that relies on the clone formation and selection during the search generations. The algorithm starts with with an initial

random population \mathbf{P} of solutions, then proceeds with the formation of clones of each individual in the population. The number of clones (C) depends upon the fitness (F) of each individual, *i.e.* better individuals have more chances to be cloned. The cloning is succeeded by hypermutation, which is similar to the mutation in Gas, the difference being that hypermutation rate \mathbf{h} , depends upon the fitness of the individuals. Thereafter, \mathbf{P} best individuals are probabilistically selected from the hypermutated clone pool C , and are again subjected to exploration moves defined as neighborhood moves, as is the case with GA (see Section 5.4.3). The moves generate better solutions and contribute to the population for the next generation. AIS based strategies are marked by a better trade off between exploration and exploitation of the search space De Castro and Von Zuben (2002). In spite of the promising and robust results, significant attempts have not been made to explore the use and applicability of AIS in the vehicle routing and scheduling problems. As a result, we incorporate AIS into the portfolio. More insights to the working of AIS can be obtained from De Castro and Von Zuben (2002).

5.4 Portfolio Evaluation and Related Performance Measures

A Portfolio is evaluated on the basis of its associated risk. The major aim to construct a portfolio is to gain computational advantage and quality while solving the given problem. The expected computational cost associated with a portfolio is the expected value of a random variable associated with the portfolio; whereas, standard deviation is the measure of the computational cost obtained while using the portfolio of algorithms. Thus, conceptually standard deviation is a measure of risk associated with the portfolio.

In this case, let $R = \{R_1, R_2, \dots, R_n\}$ be the set of random variables associated with n algorithms under consideration. Each outcome R_i is the number of function evaluations utilized to attain the prespecified performance level.

5.4.1 Mathematical & Statistical Evaluation

Let us consider that there are $\mathbf{A} = \{A_1, A_2, \dots, A_{|\mathbf{A}|}\}$ algorithms embedded in a portfolio with β processors, such that any algorithm A_i run over A_i^β processors. Hence, $\beta = A_1^\beta + A_2^\beta + \dots + A_{|\mathbf{A}|}^\beta$. As defined in the previous paragraph, a random variable R_i can be associated with an algorithm portfolio. Let R be the combined probability event distribution for all $A_i \in \mathbf{A}$. Furthermore, for an algorithm A_i it can be argued that $P[R_i]=r$ is defined as the probability that it requires the r^{th} outcome to attain the prescribed objective performance (Trivedi (2003)). It is well established for the evolutionary search techniques, that at a particular stage generating any solution has equal probability unless some pre-specified bias is not introduced (Gen and Cheng (2003)). Here, the term pre-specified bias relates to search strategies based on greedy/gradient based searches, which however, is not the case in evolutionary algorithms. Hence,

each generation in EAs can be seen as a sequence of Bernoulli trials, whereby, instead of counting the number of successes in a fixed number of trials, the number of trials is counted until the first success (obtaining best solution to a problem) is obtained, *i.e.*, attaining a specific performance level.

The fact that metaheuristics are marked by reduced search space in the later parts, the fact can be attributed to the rigorous selection pressure and generation bias. However, the proposed approach is efficient in the worst case situation, where even if it is assumed that algorithms do not perform to their search abilities and are a perfect random search, then the introduced portfolio concept is more reliable, fast, and efficient.

If a failure is denoted by 0 and a success by 1, then the sample space \mathbf{S} of the experiments consist of the set of all binary strings with an arbitrary number of 0s followed by a single 1.

$$\mathbf{S} = \{0^{r-1}1 \mid r = 1, 2, 3, \dots\} \quad (36)$$

Again, R_i is the random variable such that the value assigned to the sample point $0^{r-1}1$ is r . In order to obtain the *pmf* (Probability Mass Function) of R_i , it is evident to that the event $P[R_i] = r$ is true *if* there is a sequence of $r-1$ failures followed by one success. As is the case in the proposed approach. If the probability of each success is p , then the *pmf* of random variable R_i is given by:

$$p_{R_i}(r) = p(1-p)^{r-1}, \quad r = 1, 2, \dots, l \quad (37)$$

Where, l is the maximum number of objective function evaluations allowed to EAs.

The probability distribution function (*PDF*) of R_i , can thus be given by:

$$\begin{aligned} F_{R_i}(t) &= \sum_{r=1}^t p(1-p)^{r-1} \\ &= \frac{p[1-(1-p)^t]}{1-(1-p)} \\ &= 1-(1-p)^t \quad \text{for } t \geq 0 \end{aligned} \quad (38)$$

First, let us consider the case of two algorithms A_i and A_j , associated with random variables R_i and R_j , running on different processors to be analyzed. Let there be another random variable R , which defines the event that at least one algorithm reached the prescribed success level. Thus,

$$R = \min\{R_i, R_j\} \quad (39)$$

Let the algorithm portfolio case is investigated with β processors and n algorithms. The algorithm running over any processor i is associated with a random variable R_i , which are geometrically distributed and are mutually independent. If we let the random variable R be associated with the portfolio (set of processors and algorithms)-the following lemma holds:

Lemma 4: *The random variable R associated with the portfolio of β processors is geometrically*

distributed with parameter $1 - (1 - p)^\beta$.

Proof 4 Before going for β processors, the case with a two processor system must be analyzed. After that, it may be generalized for β processors. Therefore, the case with two processors can be defined as:

The random variable R is geometrically distributed given the two associated random variables R_i and R_j are also geometrically distributed and independent.

The probability $P(R > t)$ can be calculated provided independence ([18]); the independence of the algorithms is axiomatic as the processors considered in the paper have no interlinkage.

$$\text{Hence,} \quad P(R > t) = P(R_i > t \text{ and } R_j > t) \quad (40)$$

$$= P(R_i > t)P(R_j > t) \quad (41)$$

In terms of *PDF*,

$$\begin{aligned} P(R > t) &= 1 - F_R(t) \\ &= [1 - F_{R_i}(t)][1 - F_{R_j}(t)] \\ \Rightarrow F_R(t) &= F_{R_i}(t) + F_{R_j}(t) - F_{R_i}(t)F_{R_j}(t) \end{aligned} \quad (42)$$

From Eqn. 37,

$$\begin{aligned} F_{R_i}(t) &= \sum_{r=0}^t p(1-p)^r \\ &= 1 - (1-p)^{t+1} \end{aligned} \quad (43)$$

Similarly,

$$F_{R_j}(t) = 1 - (1-p)^{t+1} \quad (44)$$

Clearly, as per Eqn. 42,

$$\begin{aligned} F_R(t) &= 2[1 - (1-p)^{t+1}] - [1 - 2(1-p)^{t+1} + (1-p)^{2(t+1)}] \\ &= 1 - (1-p)^{2(t+1)} \\ &= 1 - [(1-p)^2]^{t+1} \end{aligned} \quad (45)$$

Hence, it can be concluded that R is also geometrically distributed with the parameter $1 - [(1-p)^2]$.

Extending the above case with two processors to β processors,

$$F_R(t) = 1 - \prod_{i=1}^{\beta} [1 - F_{R_i}(t)]$$

$$\begin{aligned}
&= 1 - \prod_{i=1}^{\beta} [1 - (1 - p)^t] \\
&= 1 - \prod_{i=1}^{\beta} [(1 - p)^t]
\end{aligned} \tag{46}$$

Thus, the random number R is geometrically distributed with parameter $1 - (1 - p)^\beta$

The following lemma states that, utilizing more than one processor in a portfolio is always a better option than relying on a single one.

Lemma 5: *The number of function evaluations for a search to end up with a desired quality solution has always more probability when used with β processors than with single processor i.e.*

$$F_{R(t)} > F_{R_i} \tag{47}$$

Proof 5: Let us consider the following relation for analysis

$$f = F_{R_i}(t) - F_R(t) \tag{48}$$

$$= (1 - (1 - p)^t) - (1 - \prod_{i=1}^{\beta} [(1 - p)^t]) \tag{49}$$

$$= \prod_{i=1}^{\beta} [(1 - p)^t] - (1 - p)^t \tag{50}$$

It can be argued that since p is the probability of success of a single trial, thus $p < 1$. Also, t and β are greater than zero. Consequentially, f can be written as

$$f = (1 - p)^t [\prod_{i=1}^{\beta} [(1 - p)^t] - 1] \tag{51}$$

Since $(1 - p)^t < 1$, therefore $\prod_{i=1}^{\beta} [(1 - p)^t]$ is bound to be less than 1, thus the first term of Eqn. 50 will be smaller. In an algorithm portfolio, more than one algorithm or processor is considered, hence, it can be concluded that in all the cases an algorithm portfolio is expected to work faster (i.e., require less objective function evaluations) than the cases with single algorithms; hence, establishing Eq. (47).

The expectation and standard deviation of a portfolio provide the measure of efficiency. Mathematically, for the case of a single algorithm with geometrically distributed random variable R_i and pmf $p_{R_i}(r) = p(1 - p)^{r-1}$, the expectation is given as:

$$\begin{aligned}
E[R_i] &= \sum_{r=1}^{\infty} r p (1 - p)^{r-1} \\
&= p \sum_{r=1}^{\infty} \frac{d}{dq} (q^r) \quad q = (1 - p) \\
&= \frac{p}{(1 - q)^2} = \frac{1}{p}
\end{aligned} \tag{52}$$

Similarly for the case of $\#$ algorithms,

$$E[R] = \frac{1}{1 - (1 - p)^\beta} \quad (53)$$

Getting away in a similar fashion for variance ' $Var'(\sigma^2)$ ' also, it can be established that:

$$Var[R_i] = \frac{(1 - p)}{p^2} \quad (54)$$

And,

$$Var[R] = \frac{1 - [1 - (1 - p)^\beta]}{[1 - (1 - p)^\beta]^2} \quad (55)$$

From the above expression, it suggests that if the probability of success and failure at any stage is constant, which can be logical in the case of random stochastic algorithms (like those considered in this study), the variance of the sample can be considered as constant. In this case, the application of Chebyshev's inequality is dianoetic, thus:

Lemma 6: *Given the non varying mean ($E[R] = \mu$) and variance ($Var[R] = \sigma^2$), the pdf of the portfolio distribution has the following bounds*

$$P(|R - \mu| \geq t) \leq \frac{\sigma^2}{t^2}, \quad t > 0. \quad (56)$$

Proof 6: *Intuitive from Chebshevs inequality [18]*

Variance can be a measure of portfolio reliability, and thus a measure of performance. Albeit the above inequality is a general upper bound, the variance is also to be assessed experimentally, and statistical analysis tools must be utilized to get a realistic outlook and to evaluate the significance of various critically connected parameters. The next subsection details the experimental design for the algorithm portfolio assessment.

5.4.2 Experimental Design

In order to obtain the best suited strategy to resolve the complex VRPSD problem, eight algorithms have been considered in this study and tested over three selective VRPSD problems of varying dimensions and complexity. Statistical analysis is performed in order to gauge the relative performance of portfolios and suggest the best portfolio under the given conditions.

In order to initiate the experiments, first all four algorithms and their corresponding enhanced neighborhood generator counterparts have been tested on the three problems under consideration. We used a single processor for 100 runs, results are reported in Figures (5-7). The Cumulative frequency illustrated in

Figures 5 to 7 are cumulative number of times an algorithm reached 0.01% range of best known objective function value. The number of function evaluations is the number of times the complex function is evaluated for obtaining the best solution. Each iteration runs according the design of portfolio and selected algorithms independently 100 times (the average results are presented). Table 1 presents the mean and variance for the number of function evaluations required by each algorithm for the three problems.

Insert Figure 5 about here

Insert Figure 6 about here

Insert Figure 7 about here

Insert Table 1 about here

Insert Table 2 about here

For the above experiment, the stopping criteria for each algorithm was set to be the performance level within 0.01% of the best known objective value or if an EA is not able to find solution that is in the range of 0.01% of best solution, then, algorithm stops when the number of function evaluations exceeds its pre-defined maximum limit. The results are presented in terms of frequency of algorithm stops within a specific range of function evaluations. Here, to gain better insight into the workings of the algorithm and to put the population based strategies and single point search strategies on the same grounds, function evaluation is taken as a performance criterion instead of a more common criterion *i.e* number of generations.

It can be clearly interpreted from Table 1, that performances of all the algorithms (excluding GA and ESA) are comparable. This empirically suggests the fact proved in equation 55. Based on the Fig. (5-7) and Table 1, it is intuitive that most of the test algorithms work in a competitive manner, thus these are the obvious choice for portfolio. The basic requirement of portfolio design is the competitive performance of the algorithms. However, keeping in view the consistent inferior performance of simple GA and enhanced SA (as evident from Fig. 5, 6 and 7) over the rest, they have not been included in further analysis, and the portfolio design has been confined to the cases of upto six algorithms. This unusual characteristic exhibited by GA can be attributed to the inefficiency due to inadequate crossover and mutation techniques. Although point based searches have experimentally performed well, the new neighborhoods did not prove to work well with an SA based search due to unavailability of elite memory and slow convergence attributes of SA.

However, the new neighborhood works well given a typical repository of elite solutions is kept, as is the case with the remaining three algorithms. Thus, the remaining experiments have been performed with various combinations of the selected six algorithms viz. enhanced GA (EGA), Simple SA, TABU search (TS), enhanced TS (ETS), simple AIS, enhanced AIS (EAIS).

Keeping in view the above mentioned performance of algorithms, they have been embedded into various portfolios, designed and analyzed in the following discussion. The tests have been performed with 2, 4, and 6 processor systems with the designed portfolios of 2, 4 and 6 algorithms. The generalized scheme to designing portfolios has been presented in Table 2.

The basis of selecting algorithms in a portfolio has been on the basis of their performance and the strategies utilized by them. For instance,

- care has been taken to give fair chances for population based strategies to compete with single solutions strategies.
- new neighborhood strategies have been kept in the portfolio with their counterpart basic algorithms.
- attempt has been made to make portfolios that also integrate single solution based and population based strategies, along with embedding basic algorithms and their enhanced counterparts in the same portfolio.

With the settings mentioned in Table 2, each portfolio has been evaluated for different combinations of selected algorithms and a '/' notation is used to denote such combinations. For example, the symbol 2/0 represents the case of 2 algorithm - 2 processor portfolio in which the first algorithm is run on both the processors and the second one is run over none. Each setting has been evaluated for 100 independent runs, and the results have been analyzed on average performance.

6. Portfolio Computing: Results, Discussions and Statistical Insights

To test the performance of the proposed methodology on the VRPSD, 3 problems of different dimensionality and complexity are simulated as discussed in Section 5.1.

6.1 Mobility Allowance

To determine the mobility allowance as described in (Eq. 30), integral approximation is carried out. The value of $E_{levy}\{\Xi\}$ was estimated to be 1.738, where $I=10^5$, $w=1$, and $l=1$. Thus, the increase in euclidean distance fraction of land paths is $\varepsilon = 1.738(eq.)$. Clearly, all the ground vehicles were assigned $\varepsilon=1.738$ fraction of any length L .

6.2 Experimental Runs

In order to initiate the experiments, all the algorithms are tuned within their parameter limits. This is done because any metaheuristic is a general algorithmic template whose parameters need to be properly tuned so that it can perform at its best. All the algorithms are tuned on a VRPSD instance to identify the parameter values at which algorithms were performing best. In case of GA, the population size is 20, crossover rate is 0.8, and mutation rate is 0.01. In case of SA, the initial temperature was set to 1000, cooling rate to be 0.98, and number of solutions checked at each stage is set to be 20. In case of AIS, the population of antibodies is 20, fraction of antibodies used for selection is 0.25, clone pool size is 40, hypermutation rate is 0.5, and fraction of low affinity antibodies to be replaced from the population in an iteration is 0.4. In case of Tabu search, exploitation factor is 10, the number of iterations that a move remains in the tabu list 16, number of iterations allowed without improvement is 300.

Based on the simulated dataset, experimentations are conducted using different portfolios as presented in this paper. The experiments are divided into 3 parts based on the number of processors used. The n-fold cross validation is conducted where $n=100$ on the different number of processors to obtain a solution quality of 0.01% of the best. The result obtained from different processors are depicted in the next sub-sections.

6.2.1 2 Processor System

For the two processor system, four selection of algorithms are analyzed as described in Table 2. Each of the four cases is analyzed for the three possible combination of the algorithms, *i.e.*, both first algorithm (A) is run on both [2/0], simultaneously or second algorithm is run on both processor[0/2]), or both algorithm can be run on different processors simultaneously as depicted in second column of Table 3. The portfolio of algorithm listed in Table 2 for 2 processor (2P) is tested for all the three problems. The results obtained for all the portfolio is shown in Figure 8.

Insert Table 3 about here

6.2.2 4 Processor System

The 2 algorithm and 4 algorithm cases are tested on a 4 independent processor system. For the 2 algorithm case, the possible combinations explored are [4/0], [3/1], [2/2], [1/3] [0/4] . The number of possible portfolios increases exponentially with an increase in processor or algorithm. In order to be concise in an exhaustive computational exercise, the authors have limited themselves with 10 possible scenarios in this study (Table 3). Figure 9 and 10 shows the results obtained for the three problems considered in this research.

Insert Figure 8 about here

Insert Figure 9 about here

6.2.3 6 Processor System

The six processor subsystem is tested with 2 algorithm, 4 algorithm and 6 algorithm cases. There are 7 cases to be explored for the 2 algorithm case. However, the large number of alternatives available with 4 algorithms, 10 cases given in Table 3 is simulated; and similarly for the 6 algorithm case 15 different scenarios are simulated. The results obtained for all the three problems have been presented in Figures 11, 12 and 13.

6.3 Results Interpretation

Based on the results obtained it is evident that the performance of the portfolios vary drastically based on the algorithm selected and number of processors used to run the algorithms. Further, as the availability for the parallel runs is limited in an organization, this adds to the compexity of portfolio selection. Due to constraints on processor and variation in algorithm performance with respect to processor subsystem selected, the selection strategy for portfolios are executed seperatly for different processor systems, *i.e.*, 2 processor, 4 processor and 6 processor cases.

To select the best portfolio for each sub-system an Analytical Hierarchy Process (AHP) based approach is utilized in this research. AHP employs hierarchical pairwise comparison to determine the weights of each alternatives. The AHP tools are used by lot of researchers in literature to determine best solution from the given set of options. A detailed study of the use of AHP can be found in Pomerol and Barba-Romero (2000).

Insert Figure 10 about here

Insert Figure 11 about here

Insert Figure 12 about here

Insert Figure 13 about here

In this research, each portfolio is recognized as an alternative for the selected processor type. The results over three different problems are considered as different attributes. First, a matrix **A** is constructed which is defined as:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

where, the columns of matrix \mathbf{A} represent attributes \mathcal{G} belong to problem type, for example, the considered Problem 1, 2 and 3, $\mathcal{G}=3$. The rows of matrix \mathbf{A} represent the alternatives (or portfolio) explored; and, a_{ij} represent the normalized function evaluation value for the experiment characterized by set (i, j) . The a_{ij} is evaluated using Eq. (57)

$$a_{ij} = \frac{FE_{ij}^{OBJ}}{\sum_j \{FE_{ij}^{OBJ}\}} \quad (57)$$

where FE_{ij}^{OBJ} is the number of function evaluations that are required by the i^{th} alternative to reach the 0.01% criterion of the j^{th} problem. Further, a_{ij} is set to zero for alternative i which is not evaluated. Thereafter, a priority matrix $\Omega_{\mathcal{G} \times \mathcal{G}}^i$ is calculated for each attribute j , where

$$\Omega_{ik}^j = \frac{a_{ij}}{a_{kj}} \quad i, k \in \mathcal{G} \quad (58)$$

The associated weight vector \mathbf{W}^j is then calculated for each attribute j by taking the geometric mean for the rows corresponding to matrix Ω^j

$$\mathbf{W}_{\mathcal{G} \times 1}^j = \{w_i^j\} \quad (59)$$

$$= \sqrt[\mathcal{G}]{\prod_k \Omega_{ik}^j} \quad (60)$$

The calculation of weight vectors are followed by their normalization, thus, three normalized priority vectors PVs are obtained as:

$$PV_i^j = \frac{w_i^j}{\sum_{j=1}^{\mathcal{G}} w_i^j} \quad (61)$$

It is clear from Fig. 14 that the above mentioned strategy can also provide a set of best alternatives if they exist. For the two processor case, the portfolio with algorithms *TS-ETS* characterized by (1/1) system is best. Similarly for the 4 processor system, the case (1/3) corresponding to the algorithm set *TS-EGA* is the best suited strategy to be followed. In the same context, for the six processor system most of the best alternative portfolios have been recognized. In these cases, top three alternatives are from the algorithm suite of *SA-TS-AIS-EAIS-EGA-ETS*. These best portfolios represent the *efficient frontier* discussed previously.

Thus, in general, it is observed that the counterparts of the algorithms that are equipped with a new neighborhood generator, work much better due to a more rigorous and focused search and their inherent characteristics of convergence possessed by them. In particular, Tabu Search based strategies can be treated as computationally more viable than other considered algorithms, as it is a part of all the best portfolios identified. It can also be observed that considering the parallel runs of the same algorithm is a better option than to choose an incoherent algorithm mix over various processors.

6.4 Validation of identified best portfolios on ten problems

To further validate the best identified portfolios on two processor, four processor and six processor systems (see Fig. 14), 10 new problems are simulated with varying size and complexity. The results corresponding to each processor system (*i.e.* maximum and minimum % deviation from best) for 100 runs is shown in Table 4. From the Table 4, it is evident that all the portfolios are able to find the solution within 2% of the best value found. Moreover, the performance of portfolio gets intensified as the number of processors are increased, which further strengthens the proposed logic of using multi processor systems. Thus, the chosen alternatives can be recommended as the best suited strategies, as they are recognized as minimum risk portfolios that competitively perform well and take care of the dimensional complexities in a better manner. Within the *efficient frontier*, a planner can choose any alternative and can also maintain flexibility in decision making as per the vision planned and other practical biases.

Insert Figure 14 about here

6.5 Discussion: Multi Processor Systems

The number of processors to be used for computation is a very influential factor in modern decision making approaches. Thus, it is very important to allocate the right number of processors associated with algorithms considered to solve problems with best possible solutions within the stipulated time frame. To understand the relation between the algorithm and number of processors needed, an exhaustive study is carried out in this research to suggest to planners, how many processors should be allocated. The AHP based approach is utilized, but with processors as alternatives, *i.e.*, 2 processor, 4 processor and 6 processor systems; and problems as attributes, *i. e.*, Problem 1, 2, and 3.

In this research, the overall variance obtained by the three aforementioned processor systems for all the test portfolios is considered as the objective function. Based on Eqs. (57) and (61), again the ranking is evaluated and the results are obtained in favour of 4 processor systems. However, before making any recommendations, it is advisable to test the cases in order to logically tap the platform dependence.

Insert Table 4 about here

7. Conclusion and Future Research

This paper presents a critical decision making combinatorial optimization problem with focus towards generating feasible and optimized solutions for vehicles moving in a stochastic environment. The paper proposes an algorithm portfolio to solve the vehicle routing problem with stochastic demand (VRPSD). The AHP based approach is utilized in this research to determine the best portfolio to get the result for a given set of processors.

The VRPSD problem considered in this research generalizes the traditional vehicle routing problem by taking into account the mobility allowance concept. Mobility allowance is incorporated in order to assimilate the real constraints imposed by the terrain, as well as to ensure more robust planning systems, and to more closely map real situations. The paper theoretically establishes the comprehensive mathematical formulation of the VRPSD problem considering mobility allowance. Furthermore, the current research presents the design and implementation of near optimal solution strategies to solve the VRSPD problem.

Since most of the decisions should be taken within a specific timeframe, the reliability and speed of the planning unit are a must. Algorithm Portfolios derive the best suited strategy that can amalgamate the superlative performance shown by various algorithms in demanding environments. One of the most enticing features is the assessment and adaptations according to the multi processor run environment. The portfolios are designed on the basis of parallel runs of various algorithms, without any interlinkage; the absence of communication help algorithms take advantage of inherent search capabilities, and the parallel implementation ameliorates of any disadvantage or randomness in the search. Also, theoretical analysis validates the claim that the chances of getting the solution faster are increased in the parallel implementation modes. Amidst the dilemma of choosing the best combination of algorithms and processors, the paper also proposes Analytical Hierarchical Process to choose the best portfolio, thus introducing a quantitative outlook to decision making. The above scheme is able to generate the *efficient frontier*, thus providing schedule planners enough scope for choosing among better alternatives, and dealing with unforeseen disturbances.

The current research focuses on performance of processors and algorithms, however, the performance of the solution is affected many tuning parameters. To make the decision making process more realistic these factors are needed to be considered in the model. Further study needs to be conducted to understand the affect of the environment with demanding response times on the performance of portfolios. The conceptualization of portfolios to combinatorial optimization is believed to make a step to resolve the dilemma of choosing the best search strategy among the nimety of the search strategies present in literature.

References

1. Agghezzaf, E.H., Raa, B., and Landeghem, H. V., Modelling inventory routing problems in supply chains of high consumption products, *European Journal of Operational Research*, 2006, 169, pp. 1048-1063.
2. Baker, B.M., and Ayechev, M.A., A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 2003, 30, pp. 787-800.
3. Bent, R. W., and Hentenryck, P. V., Scenario-Based Planning for partially Dynamic Vehicle Routing with Stochastic Customers, *Operations Research*, 2004, 52 (6), pp. 977 - 987.
4. Bernard K.-S. Cheung, K.L. Choy, Chung-Lun Li, Wenzhong Shi, Jian Tang, Dynamic routing model and solution methods for fleet management with mobile technologies, *International Journal of Production Economics*, Volume 113, Issue 2, Special Section on Advanced Modeling and Innovative Design of Supply Chain, June 2008, Pages 694-705
5. Bianchia, L., Birattarib, M., Chiarandinic, M., Manfrind, M., Mastrolillie, M., Paquetef, L., Rossi-Doriag, O., and Schiavinottoha, T., Hybrid Metaheuristicsfor the Vehicle Routing Problemwith Stochastic Demands, Technical Report No. IDSIA-06-05, IDSIA / USI-SUPSI, March 9, 2005.
6. Brandao, J., and Mercer, A., Tabu search algorithm for the multi-trip vehicle routing and scheduling problem, *European Journal of Operational Research*, 1997, 100, pp. 180-191.
7. Breedam, A.V., Improvement heuristics for the Vehicle Routing Problem based on Simulated Annealing, *European Journal of Operational Research*, 1995, 86, pp. 480-490.
8. Clarke, G. and Wright, J, Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 1964, 12, pp. 568-581.
9. De Berg, M., Gudmundsson, J., Katz, M.J., Levcopoulos, C., Overmars, M. H., and Vander Stappen, A. F., TSP with neighbourhoods of varying size, *Journal of Algorithms*, 2005, 57(1), pp. 22 - 36.
10. De Castro, L. N. & Timmis, J. I. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, London September, 357, 2002.
11. De Castro, L. N. & Von Zuben, F. J. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, 6(3), 239-251, 2002.
12. Erera, A. L., Morales, J. C., and Savelsbergh, M., The Vehicle Routing Problem with Stochastic Demand and Duration Constraints *Transportation Science*, 2010; 44(4): 474 - 492.
13. Gen, M. and Cheng, R. *Genetic Algorithm and Engineering Design*. Wiley 1997.
14. Gendreau, M. Laporte, G. and Seguin, R. Invited review: Stochastic vehicle routing *European Journal of Operational Research*, 88, 3-12, 1996.
15. Gendreau, M. Laporte, G. and Seguin, R., A tabu search heuristic for the vehicle routing problem with stochastic demands and customers, 1996, *Operations Research*, 44(3).
16. Gendreau, M. Laporte, G. and S'eguin, R. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Sciences*, 29(2), 143-155, 1995.
17. Glover, F., Laguna, M., and Marti, R., *Principles of Tabu Search, In Approximation Algorithms and Metaheuristics*. Chapman Hall/CRC, 2005.
18. Gomes, C., and Selman, B., Algorithm portfolios, *Artificial Intelligence*, 2001, 126(1-2), pp. 43-62.
19. Herer, Y. T., and Levy, R., The Metered Inventory Routing Problem, an integrative heuristic algorithm, *International Journal of Production Economics*, Volume 51, Issues 1-2, Raising the Competitive Edge in Manufacturing, 15 August 1997, Pages 69-81
20. Hong, S-C, and Park, Y-B, A heuristic for bi-objective vehicle routing with time window constraints, *International Journal of Production Economics*, Volume 62, Issue 3, 20 September 1999, Pages 249-258
21. Hwang, H-S., An integrated distribution routing model in multi-supply center system, *International*

- Journal of Production Economics, Volume 98, Issue 2, Production Research: Facing the Challenges in the New Millennium, 18 November 2005, Pages 136-142.
22. Jin, M., Liu, K., and Bowden, R.O., A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem, *International Journal of Production Economics*, Volume 105, Issue 1, January 2007, Pages 228-242.
 23. Kenyon, A. S., and Morton, D. P., Stochastic Vehicle Routing with Random Travel Times, *Transportation Science*, 2003, 37 (1), pp. 69 - 82.
 24. Kirckpatrick, S., Gelatt, Jr., C.D., and Vecchi, M.P., Optimization by simulated annealing, *Science*, 1983, 220, pp. 671-680.
 25. Lee, C.Y., Yao, X, Evolutionary Programming Using Mutations Based on the Levy Probability Distribution, *IEEE Transactions on Evolutionary Computation*, 2004, 8 (1), pp. 1-14.
 26. Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J., and Shoham, Y., A Portfolio approach to algorithm selection, In *International Joint Conference in Artificial Intelligence*, 2003, pp 1542-1543.
 27. Lim, A., and Wang, F., Multi-depot vehicle routing problem: a one-stage approach, *IEEE Transactions on Automation Science and Engineering*, 2005, 2(4), pp.397- 402.
 28. Manolis N., and Kritikos, G. I., The balanced cargo vehicle routing problem with time windows, *International Journal of Production Economics*, Volume 123, Issue 1, January 2010, Pages 42-51
 29. Mendoza, J. E., Castanier, B., Gu  ret, C., Medaglia, A. L., and Velasco, N., A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands, *Computers & Operations Research*, 37(11), 2010, pp. 1886-1898.
 30. Osman, I. H., Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of Operations Research*, 1993, 41, pp. 421-451.
 31. Park, Y-B, A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines, *International Journal of Production Economics*, Volume 73, Issue 2, 21 September 2001, Pages 175-188
 32. Pomerol, J.C., and Barba-Romero, S., *Multicriterion Decision in Management: Principles and Practice*. Kluwer Academic Publishers, Boston, 2000.
 33. Raa, B., and Aghezzaf, E. H., A robust dynamic planning strategy for lot-sizing problems with stochastic demands, *Journal of Intelligent Manufacturing*, 2005, 16(2), pp. 207-213.
 34. Ralphs, T. K. Kopman, L. Pulleyblank, W. R. and Trotter Jr., L. E. On the Capacitated Vehicle Routing Problem. *Mathematical Programming*, Ser. B 94, 343-359, Springer-Verlag, 2000.
 35. Rego, C., Node Ejection Chains for the Vehicle Routing Problem: Sequential and Parallel Algorithms, *Parallel Computing*, 2000, 27(3) pp. 201-222.
 36. Rice, J. R., The algorithm selection problem, *Advances in Computers*, 1976, 15, pp. 65-118.
 37. Secomandi, N., Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands, *Computers and Operations Research*, 2000, 27(5), pp. 1171-1200.
 38. Stewart, W. and Golden, B., Stochastic vehicle routing: comprehensive approach. *European Journal of Operational Research*, 1983, 14, pp. 371-385.
 39. Stinson, D. R., An Introduction to the Design and Analysis of Algorithms, *The Charles Babbage Research Centre, Winnipeg, Manitoba, Canada, 2nd Edition*, 1987.
 40. Sungur, I., Ord   ez, F., and Dessouky, M.M, A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty, *IIE Transactions* 40 (5), 2008, pp. 509-523.
 41. Teodorovi  c, D. and Pavkovi  c, G., A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand, *Transportation Planning and Technology*, 1992, 16, 261-273.
 42. Trivedi, K. S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons Inc., 2003

43. Wasner, M., and Zapfel, An integrated multi-depot hub-location vehicle routing model for network planning of parcel service, *International Journal of Production Economics*, Volume 90, Issue 3, Production Control and Scheduling, 18 August 2004, Pages 403-419.
44. Winkler, R. L., *Decision Modeling and Rational Choice: AHP and Utility Theory*, Management Science, 1990, Vol. 36, pp. 247-248.
45. Xiangyong, L., Peng, T., and Leung, S.C.H., Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm, *International Journal of Production Economics*, Volume 125, Issue 1, May 2010, pp. 137-145
46. Yang, W., Mathur, K., and Ballou, R. H. Stochastic vehicle routing problem with restocking. *Transportation Science*, 2000, 34(1), pp. 99–112.
47. Zapfel, G., and Bogl, M., Multi-period vehicle routing and crew scheduling with outsourcing options, *International Journal of Production Economics*, Volume 113, Issue 2, Special Section on Advanced Modeling and Innovative Design of Supply Chain, June 2008, pp. 980-996.
48. Shukla, N., Tiwari, M.K., Ceglarek, D., 2012 "Genetic-algorithms-based algorithm portfolio for inventory routing problem with stochastic demand." *International Journal of Production Research*, DOI:10.1080/00207543.2011.653010.
49. Brown, K.L., Nudelman, Eugene, Galen, Andrew, McFadden, J., Shoham, Y., "A Portfolio approach to algorithm selection", *Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, August 9-15, 2003.
50. Moghaddama, B. F., Ruizb, R., Sadjadic, S.J. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm" *Computers & Industrial Engineering*, Volume 62, Issue 1, 2012, pp. 306–317.
51. Goodson, J., J. Ohlmann, and B. Thomas, Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. Forthcoming in *European Journal of Operational Research*, 2012

Table 1: Mean and variance of the function evaluations for the three problems

Algorithm	Problem 1		Problem 2		Problem 3	
	Mean	Variance	Mean	Variance	Mean	Variance
GA	1.70	1.03	2.46	0.98	3.03	1.61
EGA	0.96	0.58	2.13	0.60	2.59	1.50
SA	1.45	1.09	2.26	0.70	2.44	1.07
ESA	1.85	1.49	2.64	1.47	2.72	2.07
AIS	1.43	0.63	2.00	0.63	2.39	1.83
EAIS	1.36	0.64	1.97	0.66	2.20	1.71
TABU	1.40	0.56	1.86	0.47	2.19	1.22
ETABU	0.85	0.49	1.64	0.58	2.03	1.09

* Mean and variance is scaled at problem level

Table 2: Portfolio Design Scheme

No of algorithms	2 Processors			4 Processors					6 Processors						
2	2P-2A	S1	A2	4P-2A	S1	A2			6P-2A	S1	A2	-	-	-	-
		T1	T2		T1	T2	-	-		T1	T2	-		-	-
		A1	G2		A1	G2	-	-		A1	G2	-	-	-	-
		T2	G2		T2	G2	-	-		T2	G2	-	-	-	-
4	N/A	-	-	4P-4A	T1	T2	S1	A2	6P-4A	T1	T2	S1	A2	-	-
		-	-		A1	G2	A2	T2		A1	G2	A2	T2	-	-
6	N/A	-	-	N/A	-	-	-	-	6P-6A	S1	T1	A1	A2	G2	T2

G2=EGA; S1=SA; A1=AIS; A2=EAIS; T1=TS; T2=ETS

*N/A – Not applicable as number of processors is less than number of algorithms to run

Table 3: Cases investigated with processor to algorithm system

S.No.	2P-2A	4P-2A	4P-4A	6P-6A	6P-4A	6P-2A
1	2/0	4/0	4/0/0/0	6/0/0/0/0/0	6/0/0/0	6/0
2	1/1	3/1	0/4/0/0	0/6/0/0/0/0	0/6/0/0	5/1
3	0/2	2/2	0/0/4/0	0/0/6/0/0/0	0/0/6/0	4/2
4	--	1/3	0/0/0/4	0/0/0/6/0/0	0/0/0/6	3/3
5	--	0/4	3/1/0/0	0/0/0/0/6/0	1/2/2/1	2/4
6	--	--	3/0/1/0	0/0/0/0/0/6	2/2/2/0	1/5
7	--	--	3/0/0/1	1/1/1/1/1/1	0/2/2/2	0/6
8	--	--	2/1/1/0	2/2/2/0/0/0	1/1/2/2	--
9	--	--	2/0/1/1	0/0/0/2/2/2	4/2/0/0	--
10	--	--	1/1/1/1	0/2/2/2/0/0	3/0/0/3	--
11	--	--	--	0/0/2/2/2/0	2/1/1/2	--
12	--	--	--	2/0/2/0/2/0	0/0/1/5	--
13	--	--	--	0/2/0/2/0/2	5/1/0/0	--
14	--	--	--	1/2/1/1/0/1	3/1/1/1	--
15	--	--	--	1/1/2/1/1/0	1/1/1/3	--

2P-2A: 2 Processor – 2 Algorithm, 4P-2A: 4 Processor-2 Algorithm, 4P-4A: 4 Processor - 4 Algorithm; 6P-6A: 6 Processor- 6 Algorithm; 6P-4A: 6 Processor - 4 Algorithm; 6P-2A: 6 Processor - 2 Algorithm

Table 4: Results obtained by running best processors on ten randomly generated problems

Problem No	Customers	Vehicles	2P System		4P System		6P System	
			% deviation from best					
			Min	Max	Min	Max	Min	Max
1	20	2	0.00	0.11	0.00	0.08	0.00	0.01
2	40	4	0.00	0.20	0.00	0.19	0.00	0.17
3	60	6	0.01	0.39	0.00	0.39	0.00	0.26
4	80	8	0.02	0.52	0.01	0.48	0.01	0.44
5	100	10	0.03	0.76	0.03	0.68	0.09	0.58
6	120	12	0.07	0.88	0.09	0.72	0.07	0.63
7	140	14	0.08	1.02	0.12	0.80	0.11	0.79
8	160	16	0.19	1.16	0.13	0.93	0.10	0.82
9	180	18	0.35	1.37	0.22	1.19	0.16	1.12
10	200	20	0.33	1.88	0.26	1.72	0.13	1.36

2P denotes portfolio T1-T2 with two processors;

4P denotes portfolio T1-G2 with four processors;

6P denotes portfolio S1-T1-A1-A2-G2-T2 with six processors

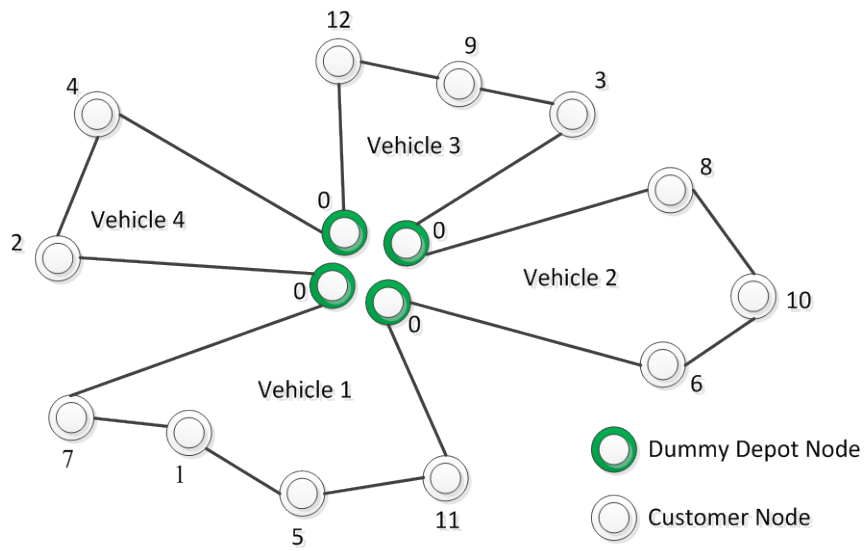


Figure 1: Cyclic TSP string representing four vehicle priori tours

```

For ( $k = 1, 2, 3, \dots, TNV$ ) do
  For ( $\lambda = \Lambda, \Lambda - 1, \Lambda - 2, \dots, 0$ ) do
     $f_{n_k}(\lambda) = c_{n_k,0}$ 
    For ( $j = n_k - 1, n_k - 2, \dots, 1$ ) do
      //  $n_k$  is the last customer visited by the  $k^{th}$  vehicle
      Compute  $f_{jk}^r(\lambda)$  using  $f_{(j+1)k}(\cdot)$  (using Eqn. 3)
      For ( $\lambda = \Lambda, \Lambda - 1, \Lambda - 2, \dots, 0$ ) do
        Compute  $f_{jk}^p(\lambda)$  (using Eqn. 2)
        Compute  $f_{jk}^r(\lambda)$  and  $f_{jk}^p(\lambda)$  for finding threshold
         $LT_{jk}$  (using Eq. 3)
        Compute  $f_{jk}(\lambda)$  using  $f_{(j+1)k}(\lambda)$  (using Eqn. 1)
      End For
    End For
  End For
  Compute  $f_{0k}(\Lambda)$ 
   $OBJ = OBJ + f_{0k}(\Lambda)$ 
End For
Return ( $OBJ$ )

```

Figure 2: Procedure for Computation of the VRPSD objective function

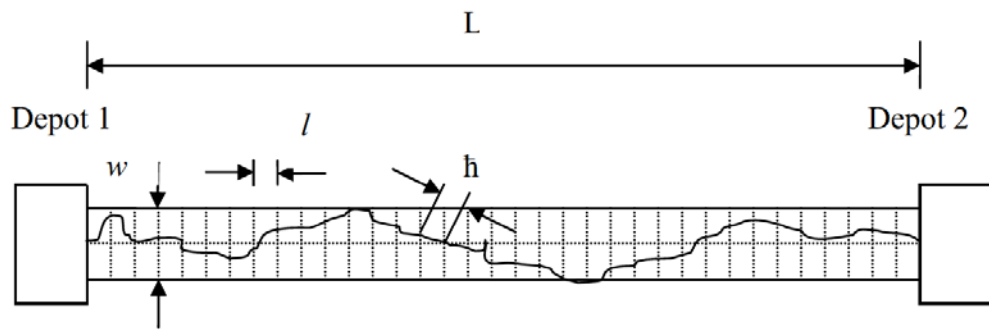


Figure 3: Conceptualized movement of vehicle with mobility allowance

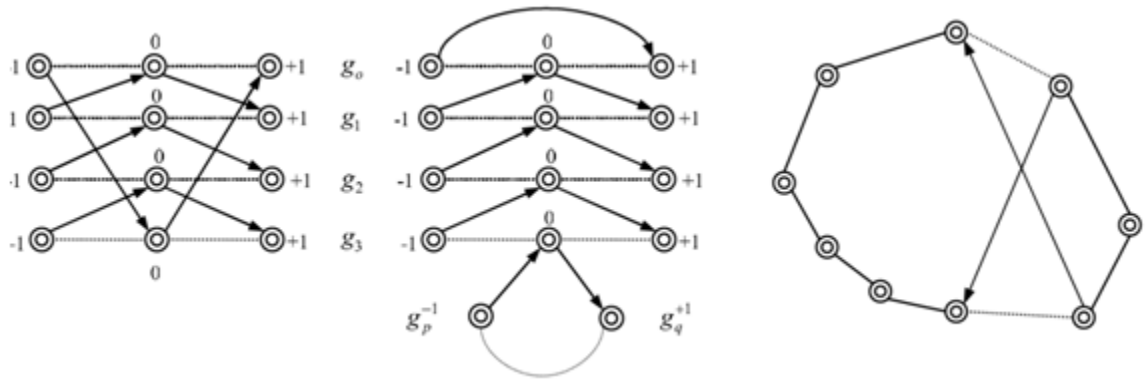


Figure 4: Three different neighborhood search approaches (in this figure, dark and faded lines between the nodes denote the links that are formed after and before the transformation \mathbf{T})

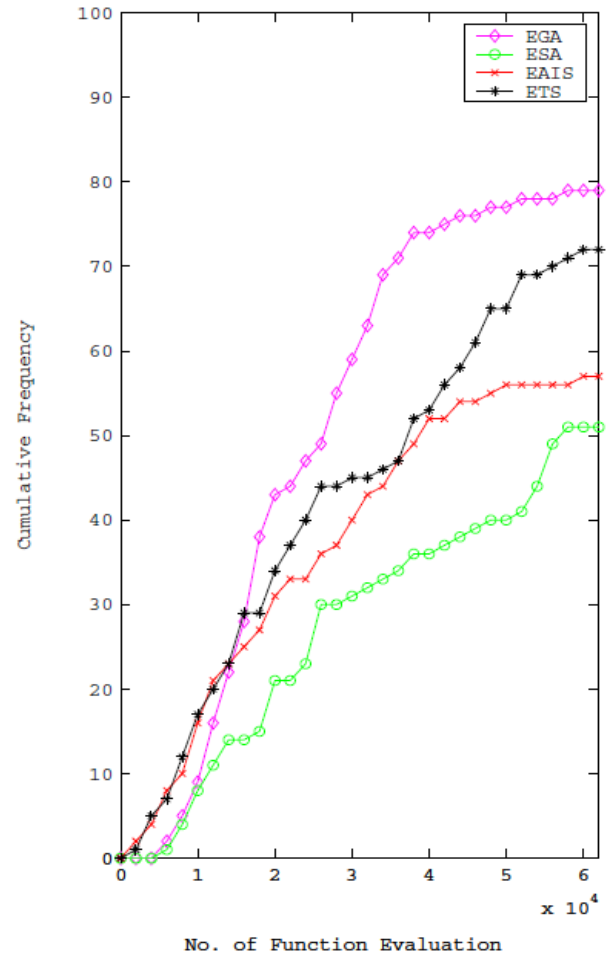
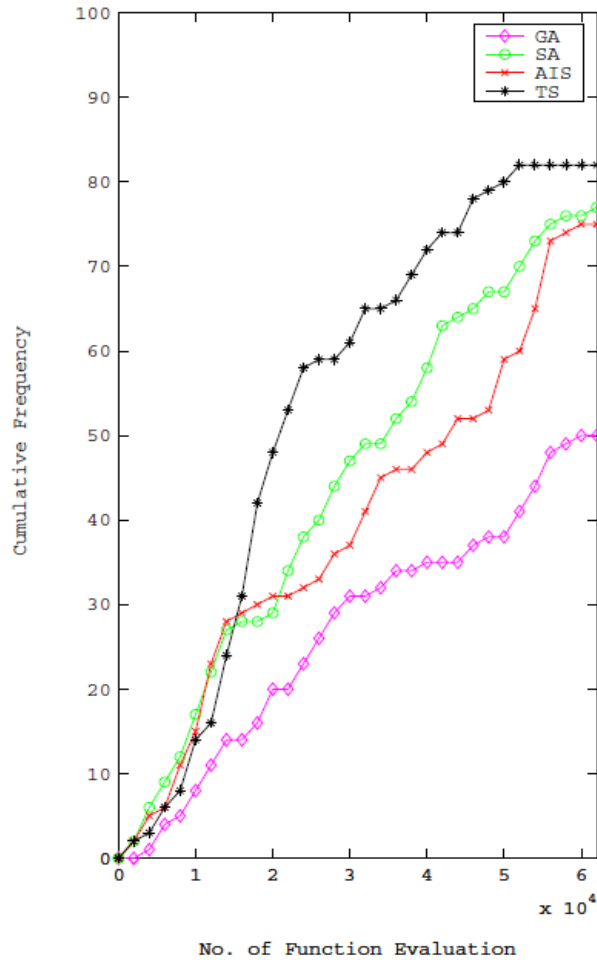


Figure 5: Cumulative frequency plots of EAs for small instance of VRPSD with mobility allowance as described in Section 4.1; Cumulative frequency illustrated in graphs are cumulative number of times an EA reached 0.01% range of best known objective function value

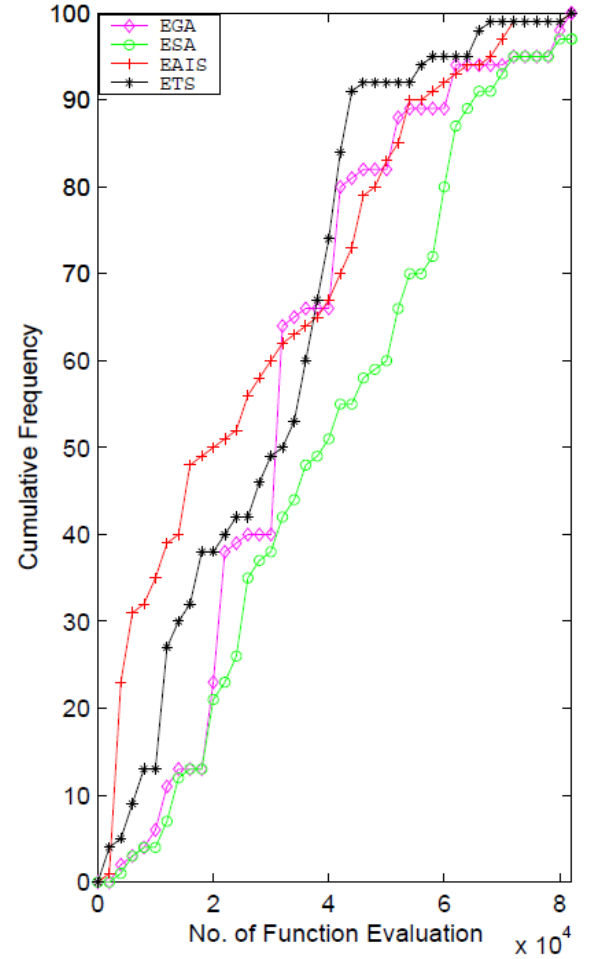
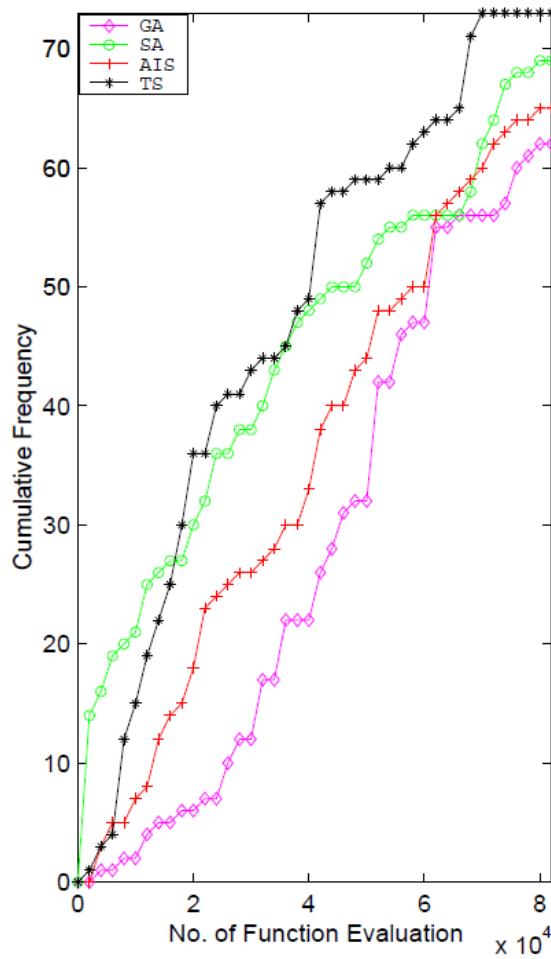


Figure 6: Cumulative frequency plots of EAs for medium instance of VRPSD with mobility allowance as described in Section 4.1; Cumulative frequency illustrated in graphs are cumulative number of times an EA reached 0.01% range of best known objective function value

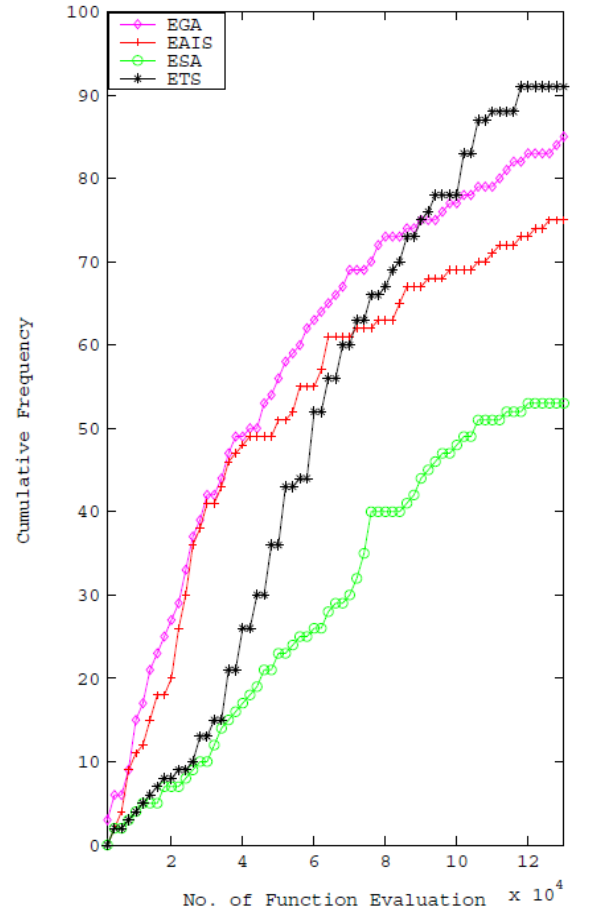
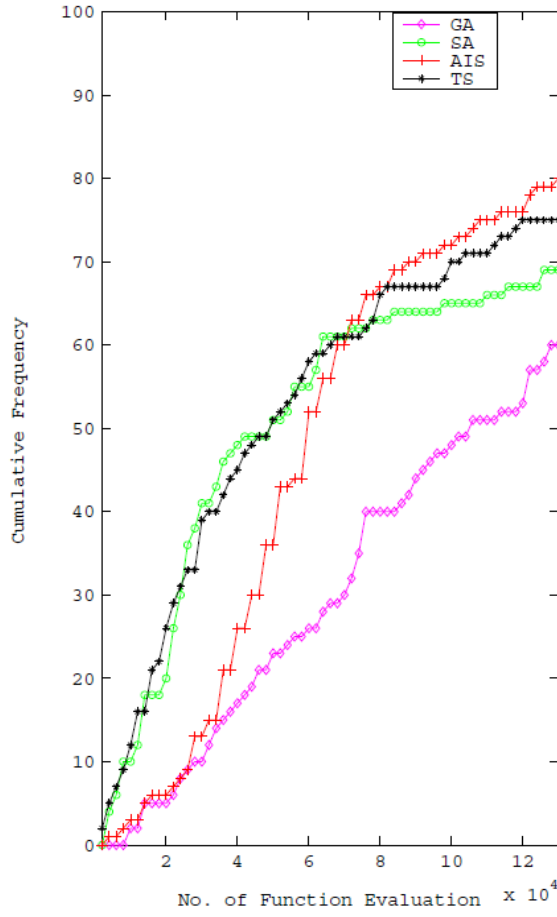


Figure 7: Cumulative frequency plots of EAs for large instance of VRPSD with mobility allowance as described in Section 4.1; Cumulative frequency illustrated in graphs are cumulative number of times an EA reached 0.01% range of best known objective function value

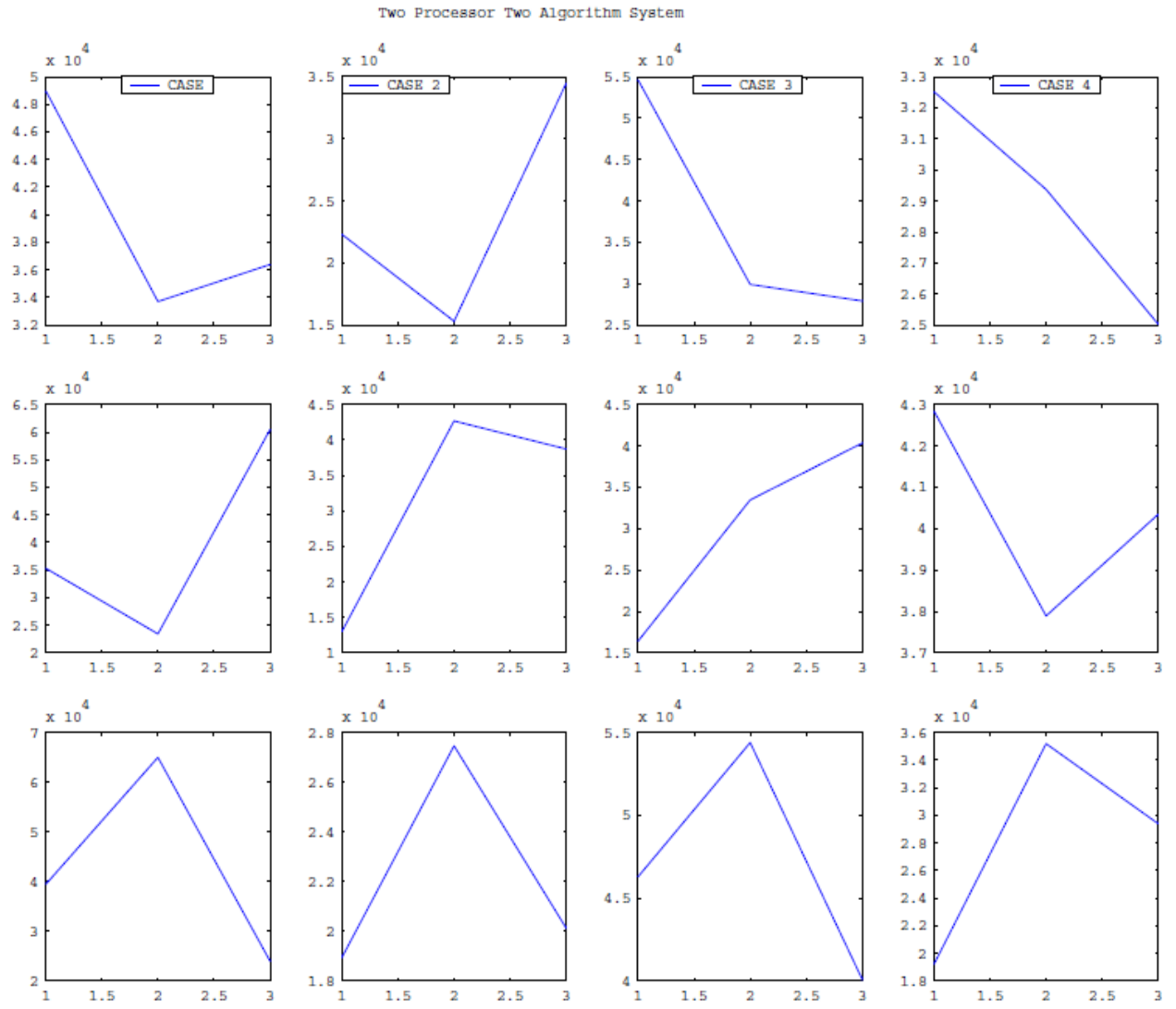


Figure 8: Results for different portfolio cases pertaining to two algorithm-two processor system. The three rows present results for the three problems considered and the columns are corresponding to different algorithms selected. Y axis is scaled to number of function evaluations; X axis denotes the portfolio cases explored

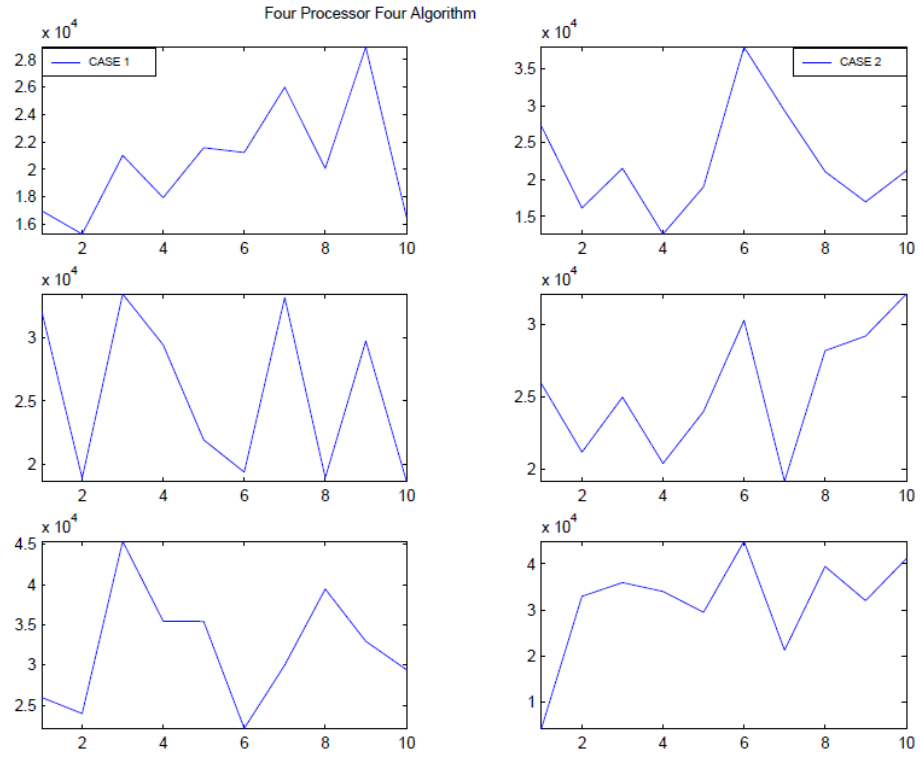


Figure 9: Results for different portfolio cases pertaining to four algorithm-four processor system. The three rows present results for the three problems considered and the columns are corresponding to different algorithms selected. Y axis is scaled to number of function evaluations; X axis denotes the portfolio cases explored

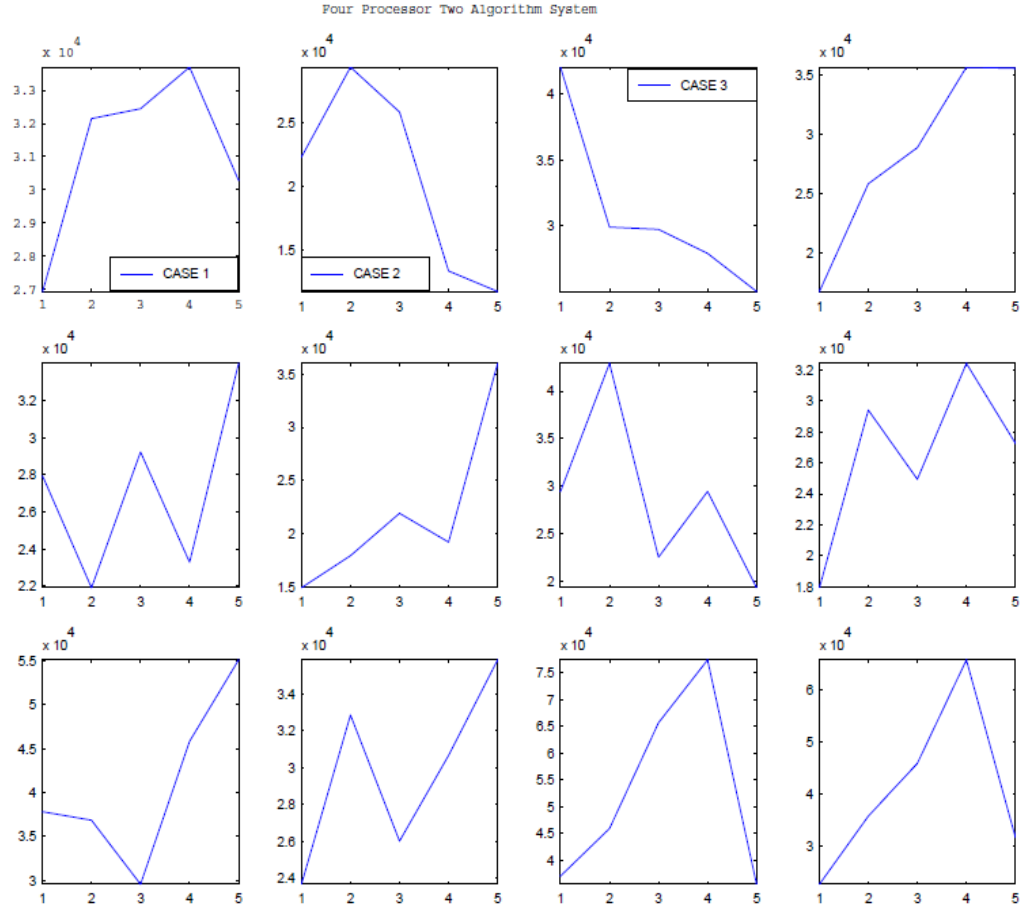


Figure 10: Results for different portfolio cases pertaining to two algorithm-four processor system. The three rows present results for the three problems considered. Y axis is scaled to number of function evaluations; X axis denotes the portfolio cases explored

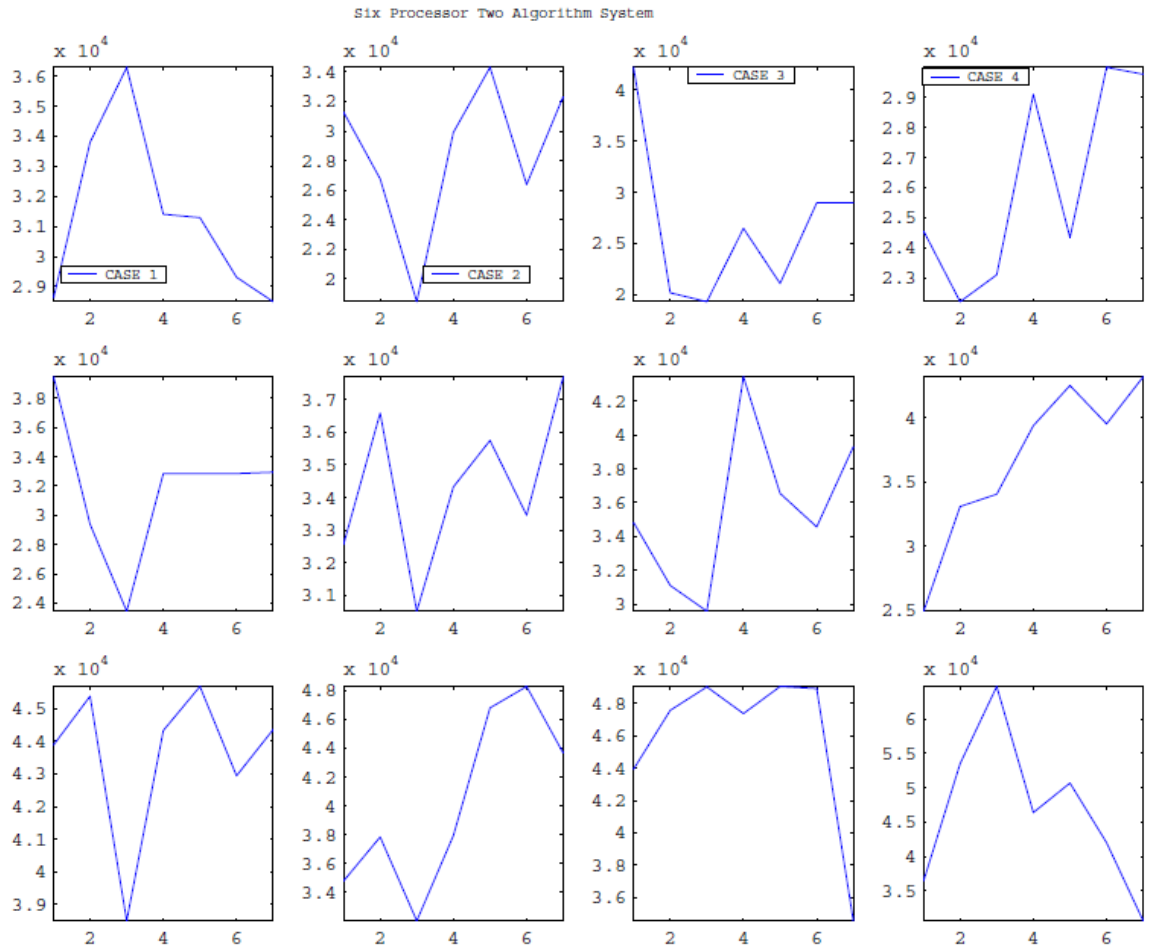


Figure 11: Results for different portfolio cases pertaining to six algorithm-two processor system.

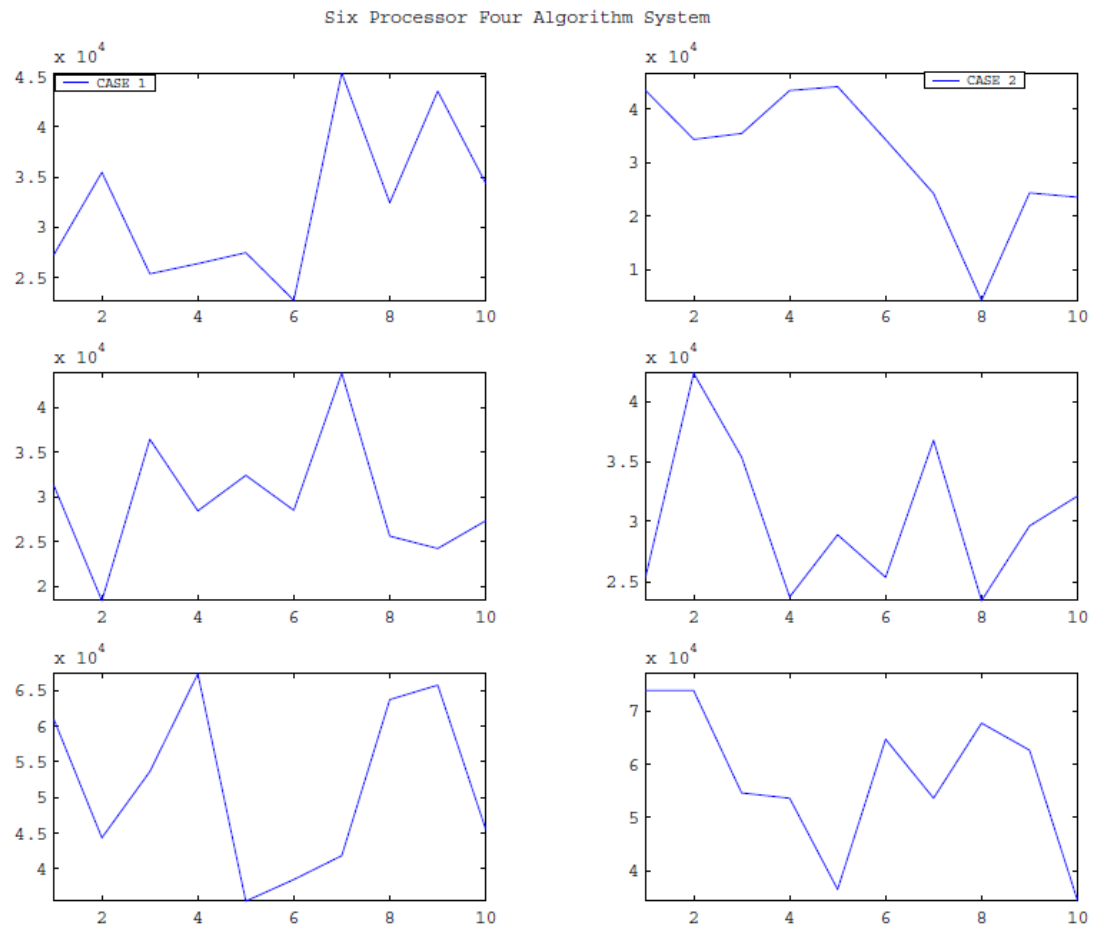


Figure 12: Results for different portfolio cases pertaining to six algorithm-four processor system.

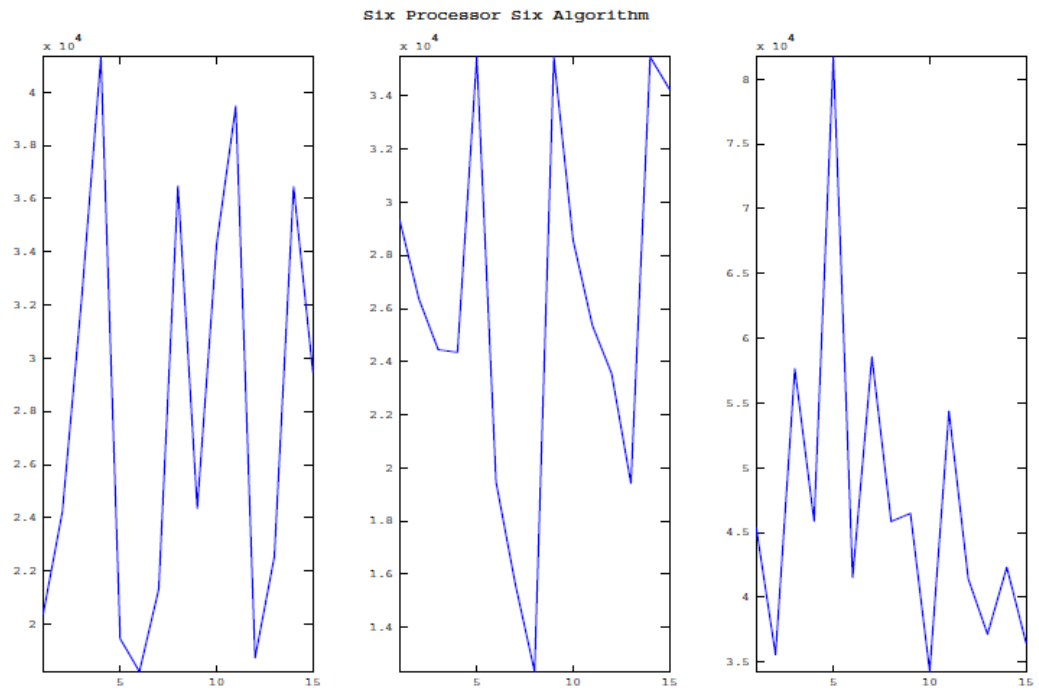


Figure 13: Results for different portfolio cases pertaining to six processor-six algorithm system.

TWO PROCESSOR SYSTEM			
Priority Vectors	Ranks	Processor Cases	Algorithms in Portfolio
0.695	12	2/0	S1/A2
0.668	10	1/1	
0.678	11	0/2	
0.469	2	2/0	T1/T2
0.304	1	1/1	
0.527	4	0/2	
0.661	0	2/0	A1/G2
0.645	8	1/1	
0.59	7	0/2	
0.533	5	2/0	T1/G2
0.568	6	1/1	
0.525	3	0/2	

FOUR PROCESSOR SYSTEM			
Priority Vectors	Ranks	Processor Cases	Algorithms in Portfolio
0.5932	26	4/0	S1/A2
0.5835	23	3/1	
0.6113	29	2/2	
0.6453	33	1/3	T1/T2
0.7412	37	0/4	
0.3947	4	4/0	
0.5136	18	3/1	A1/G2
0.487	13	2/2	
0.3872	3	1/3	
0.5272	21	0/4	T1/G2
0.7207	35	4/0	
0.7676	38	3/1	
0.6929	34	2/2	A1/G2
0.783	39	1/3	
0.5007	17	0/4	
0.8182	40	4/0	T1/G2
0.5867	24	3/1	
0.6201	30	2/2	
0.3691	1	1/3	T1/T2/S1/A2
0.6297	32	0/4	
0.4956	15	4/0/0/0	
0.3715	2	0/4/0/0	T1/T2/S1/A2
0.6214	31	0/0/4/0	
0.5232	20	0/0/0/4	
0.494	14	1/1/1/1	T1/T2/S1/A2
0.4143	7	2/2/0/0	
0.5923	25	0/0/2/2	
0.4764	12	0/2/2/0	A1/G2/A2/T2
0.6018	28	2/0/0/2	
0.4029	5	2/1/1/0	
0.4359	9	4/0/0/0	A1/G2/A2/T2
0.4343	8	0/4/0/0	
0.5193	19	0/0/4/0	
0.4047	6	0/0/0/4	A1/G2/A2/T2
0.4639	10	1/1/1/1	
0.73	36	2/2/0/0	
0.473	11	0/0/2/2	A1/G2/A2/T2
0.556	22	0/2/2/0	
0.4991	16	2/0/0/2	
0.5944	27	2/1/1/0	A1/G2/A2/T2

SIX PROCESSOR SYSTEM			
Priority Vectors	Ranks	Processor Cases	Algorithms in Portfolio
0.6891	42	6/0	S1/A2
0.6566	31	5/1	
0.602	17	4/2	
0.6611	34	3/3	T1/T2
0.6657	35	2/4	
0.64	26	1/5	
0.6405	27	0/6	A1/G2
0.6191	19	6/0	
0.6289	21	5/1	
0.498	5	4/2	T1/G2
0.6352	24	3/3	
0.7146	53	2/4	
0.6449	28	1/5	A1/G2
0.7021	47	0/6	
0.7537	58	6/0	
0.5787	14	5/1	T1/T2
0.5669	13	4/2	
0.7174	54	3/3	
0.6329	23	2/4	A1/G2
0.6749	36	1/5	
0.6523	30	0/6	
0.5187	7	6/0	T1/G2
0.6328	22	5/1	
0.6925	44	4/2	
0.7024	48	3/3	T1/G2
0.7087	50	2/4	
0.6922	43	1/5	
0.6868	41	0/6	T1/T2/S1/A2
0.5814	15	6/0/0/0	
0.6822	39	0/6/0/0	
0.6848	40	0/0/6/0	T1/T2/S1/A2
0.5928	16	0/0/0/6	
0.5407	9	1/2/2/1	
0.8373	61	2/2/2/0	T1/T2/S1/A2
0.693	45	0/2/2/2	
0.7723	59	1/1/2/2/	
0.6458	29	4/2/0/0	A1/G2/A2/T2
0.8142	60	3/0/0/3	
0.8754	62	6/0/0/0	
0.7521	57	0/6/0/0	A1/G2/A2/T2
0.719	55	0/0/6/0	
0.6937	46	0/0/0/6	
0.7091	51	1/2/2/1	A1/G2/A2/T2
0.6762	38	2/2/2/0	
0.4856	4	0/2/2/2	
0.6596	33	1/1/2/2/	A1/G2/A2/T2
0.5572	11	4/2/0/0	
0.5574	12	3/0/0/3	
0.5233	8	6/0/0/0/0/0	S1/T1/A1/A2/G2/T2
0.6577	32	0/6/0/0/0/0	
0.6757	37	0/0/6/0/0/0	
0.7457	56	0/0/0/6/0/0	S1/T1/A1/A2/G2/T2
0.4512	1	0/0/0/0/6/0	
0.5143	6	0/0/0/0/0/6	
0.5488	10	1/1/1/1/1/1	S1/T1/A1/A2/G2/T2
0.6379	25	2/0/2/0/2/0	
0.6084	18	0/2/0/2/0/2	
0.7045	49	1/1/2/0/1/1	S1/T1/A1/A2/G2/T2
0.4855	3	2/2/2/0/0/0	
0.4646	2	0/0/0/2/2/2	
0.7098	52	1/2/1/0/0/2	S1/T1/A1/A2/G2/T2
0.6256	20	3/0/3/0/0/0	

Figure 14: Algorithm Portfolio performance evaluation using AHP