1-1-2008

Introducing algorithm portfolios to a class of vehicle routing and scheduling problem

Nagesh Shukla
National Institute of Foundry and Forge Technology, nshukla@uow.edu.au

Yogesh Dashora
University of Texas

M K. Tiwari
National Institute Of Foundry And Forge Technology

F T. S Chan
University of Hong Kong

T C. Wong
University of Hong Kong

Introducing algorithm portfolios to a class of vehicle routing and scheduling problem

Abstract

The paper presents a comprehensive foundation and implementation of Algorithm Portfolios to solve Theater Distribution Vehicle Routing and Scheduling Problems (TDVRSP). In order to evaluate the performance of proposed approach, it has been applied to varying dimensions of theater distribution problem. In particular, eight random search metaheuristics embedded in four processors, packed to form different portfolios. Four basic algorithms- Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and Artificial Immune System (AIS), as well as their group theoretic counterparts have been utilized. The proposed approach also takes care of platform dependence and helps evolving a robust solution pack. The portfolio concept is shown to be computationally advantageous and qualitatively competitive over the benchmark set of problems. The paper does not only provide modeling to TDVRSP, but also aids in developing a generic solution framework for other problems of its kind.

Keywords

vehicle, introducing, class, portfolios, problem, scheduling, routing, algorithm

Publication Details

Shukla, N., Dashora, Y., Tiwari, M. K., Chan, F. T. S. & Wong, T. C. (2008). Introducing algorithm portfolios to a class of vehicle routing and scheduling problem. Proceedings of The 2nd International Conference on Operations and Supply Chain Management (pp. 1-10).

INTRODUCING ALGORITHM PORTFOLIOS TO A CLASS OF VEHICLE ROUTING AND SCHEDULING PROBLEM

Nagesh Shukla[1], Yogesh Dashora[2], M K Tiwari[3], F T S Chan[4], and T C Wong[5]
[1]Department of Manufacturing Engineering,
National Institute of Foundry and Forge Technology, Ranchi INDIA 834003
Email: nagesh.nifft@gmail.com
[2]Operations Research & Industrial Engineering, Mechanical Engineering Department
University of Texas, Austin USA TX 78712-0292 USA
Email: yogesh.dashora@mail.utexas.edu
[3]Department of Forge Technology
National Institute of Foundry and Forge Technology, Ranchi INDIA 834003
Tel: +91651-2291116, Fax: +91651-2290860
Email: mkt09@hotmail.com
[4]Department of Industrial and Manufacturing Systems Engineering
University of Hong Kong, Pokfulam Road Hong Kong
Tel: +852 2859 7059, Fax: +852 2858 6535
Email: ftschan@hkucc.hku.hk
[5]Department of Industrial and Manufacturing Systems Engineering
University of Hong Kong, Pokfulam Road Hong Kong
Email: h9901225@hkusua.hku.hk

ABSTRACT
The paper presents a comprehensive foundation and implementation of Algorithm Portfolios to solve Theater Distribution Vehicle Routing and Scheduling Problems (TDVRSP). In order to evaluate the performance of proposed approach, it has been applied to varying dimensions of theater distribution problem. In particular, eight random search metaheuristics embedded in four processors, packed to form different portfolios. Four basic algorithms- Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and Artificial Immune System (AIS), as well as their group theoretic counterparts have been utilized. The proposed approach also takes care of platform dependence and helps evolving a robust solution pack. The portfolio concept is shown to be computationally advantageous and qualitatively competitive over the benchmark set of problems. The paper does not only provide modeling to TDVRSP, but also aids in developing a generic solution framework for other problems of its kind.

KEY WORDS
Metaheuristics, Vehicle Routing and Scheduling Problem

1. Introduction

The paper introduces the algorithm portfolios concept to resolve a class of vehicle routing and scheduling problem known as theater distribution vehicle routing and scheduling problem (TDVRSP). A basic Theater Distribution Vehicle Routing and Scheduling Problem (TDVRSP) deals with finding an economically efficient and time-definite delivery of the flow of personnel and material within the theater in order to fulfill the desired aims [1][2]. None of the early researches did focus on optimizing and prescribing the routing and scheduling plans for the utilization of assets in the theater while satisfying the customer demands and time requirements. Initially, such an optimization attempt was introduced by Crino [2] in a pioneering dissertation work and was well established in the literature by [3].

Keeping the fact that even a simple vehicle routing problem with scheduling objectives is *NP* hard [6], the additional constraints imposed due to consideration of theater distribution scenario makes the problem more complex. The optimal strategies do find their application to academic toy problems of insignificant dimensions, but the real dimensions demand more robust heuristic and metaheuristic approximation approaches so that better decisions, if not optimal, can be made within the stipulated time frames. The increasing use of metaheristics has spectacularly reduced the time of response without much depreciation in terms of solution quality. In particular, the state of the art random evolutionary metaheuristics like Genetic Algorithm (GA) [9], Tabu Search (TS) [7], Simulated Annealing (SA) [10], and more recent Artificial Immune System (AIS) [8] have been extensively used and their robustness and capability to solve complex combinatorial real time problems is well established. There have also been successful instances of amalgamation of Group Theory with metaheuristics, in particular, TS for solving the underlying TDVRSP [2][3]. Till now the research focus was on to get a feasible and considerably good solution for the extremely complex TDVRSP; the upcoming research focus should be the quality of solution and response time of the methodology adopted. Hence, one of the focuses of the present paper is to introduce algorithm portfolio for the underlying TDVRSP; analyze the relative performance of various strategies embedded in the portfolio and get an insight to the solution quality with the increasing problem complexity.

A portfolio of algorithms can be defined as *A collection of different algorithms and/or different copies of the same algorithm running on different processors* [4][5]. The need of portfolio for the complex combinatorial problems arise from the fact that neither any random search algorithm can guarantee to be the best suited strategy for a particular type of problem, nor do the algorithm performance remain similar for varying dimensions and complexities of the same problem. The importance of the present paper lies in introducing algorithm portfolios, *for the first time*, using the random search metaheuristics to solve complex real dimension TDVRSP. Keeping in view the success of group theory in solving the TDVRSP [3], the proposed portfolios integrate the aforementioned four basic evolutionary metaheuristics and their corresponding group theoretic counterparts, thus ensuring a more robust experimental ground.

The remainder of the paper is organized as follows: Section 2 formally describes the problem statement; section 3 describes the algorithm portfolio concept, details the implementation aspects of four basic algorithms, and describes the experimental design utilized; section 4 develops with critical insights to the comparative performance of the algorithms and analyzes functioning of the portfolios along with suggestions to choosing best portfolios; and finally, section 5 concludes the paper with suggested future research.

2. Problem Statement

A theater distribution is defined as a flow of material, personnel and equipment within a theater to suit the intents of combatant commander. Theater distribution network nodes are depots, hubs and customers (for more information see [2]). Depots, aerial ports of debarkation (APODs), and seaports of debarkation (SPODs) are the supply nodes. Customers are the sink nodes that receive cargo. All the nodes have their own time window constraints, fuel storage capacity and maximum on ground (MOG) constraints. Hubs have cargo storage constraints and customers have cargo demand requirements and time definite delivery requirements.
The TDVRSP has tiered distribution architecture. The first order tier contains the depots and customers/hubs served by the depots. Middle tiers consist of hubs that service customers/hubs. The last order tier consists of end customers served by a hub. Each tier is a self-contained distribution network.

The primary objectives of TDVRSP are to minimize unmet customer demand defined as demand filled shortfall (DFS), late deliveries called Time Definite Delivery shortfall (TDDS), vehicle fixed costs (FC), and vehicle variable costs (VC). The demand shortfall is the difference between the customer's total demand and the sum total amount of demand delivered to the customer. The TDDS is the weighted difference between the customer's desired delivery time and late delivery time for a set amount of demand. Fixed cost is the total cost for all vehicles used to deliver goods and services. Variable costs are the total costs for vehicles to travel the prescribed routes. Penalty costs include parking Maximum on Ground violations and hub storage capacity violations (MOGP). Storage Penalty (SP) refers to the penalty added when the storage at hubs exceeds the predefined limits. More insights to the problem can be found in [2]. The objective function (OBJ) formulated consists of sum of all the above mentioned penalties, and is shown below:

$$OBJ = DFS + TDDS + FC + VC + MOGP + SP \qquad (1)$$

3. Portfolio Design

3.1 Motivation and Instances Explored

Albeit some algorithms usually perform better than others on average, it is yet not possible to define a best algorithm for the given problem/problem set. Rather, different problem instances are usually associated with varying run time complexities [11]. Mostly, the algorithm selection was attempted using ``winner take all" strategy, *i.e.* select a best performing algorithm based on its performance over the given problem in various attempts [12].

Table 1: Benchmark Characteristics

Problem No.	Size	Del. Restrictions	Dem./Cap. Ratio
1	AF-MTMS - small	Low	Low
14	J-MTMS-Medium	Medium	Medium
23	J-MTMS w/hub	Medium	Medium

The literature pertaining to TDVRSP is not very rich and only a few remarkable contributions exist in this area [3]. In addition, only the dissertation by Crino [2] explicitly details the benchmark data sets used for the study of robust performance of Group Theoretic Tabu search (GTTS). Problem sets are characterized in low, medium and high level hierarchy depending upon problem size, problem density, demand/capacity ratio, and other defining constraints. The three TDVRSP types characterized in the research were: Air Force Multiple trips multiple services (MTMS) without hub, Joint MTMS without hub, joint MTMS with hub and other defining constraints. In this study one problem from each category has been attempted *i.e.*, Problem Number 1, 14, and 23, which are hereafter referred as Problem 1, 2 and 3, respectively. Such a selection of problems was intuitive based on the problem characteristics as described in Table 1.

The actual data set used for the problem instances can be found in [2]. The solution to these problems has been attempted by integrating them in a algorithm portfolio that works on the strategy of minimizing the risk in terms of computational cost and the solution quality obtained.

3.2 The Portfolio and Algorithmic Suit

Portfolio Design basically targets to minimize the expected risk and gain 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 in the upcoming subsections. 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 the present case, the paper utilizes portfolios consisting of eight algorithms or more specifically, four metaheuristics and their group theoretic counterparts. The earlier attempts to solve TDVRSP were concerned to using group theoretic Tabu search [2], however, an obvious choice is to search for other strategies as well. But, there are circumstances where one needs to just find good solution quickly rather than getting best solution always, thereby intensifying the need for best portfolios with all required adaptabilities.

All the four algorithms require similar representation schema, although the representation changes with group theoretic counterparts. In the interest of brevity and to avoid rephrasing the literature, the following discussion presents a crisp overview of the implementation of algorithms. First, the encoding schema is detailed as follows:

3.2.1 Encoding Schema

Basic Representation: A representative solution to the proposed TDVRSP consist of the string that contains customer numbers C based on total services allowed and vehicle numbers V based on the total vehicle trips, such that any element of $|C| > |V|$. The string also contains the partition variable '$\#$' that is used to distinguish one tier from the other. The strings/substrings before or after '$\#$' can be treated as independent to one another.
Group theoretic representation: The application of group theoretic aspects to the algorithms detailed above require the representation of solution as elements of *symmetric group on n-letters*, S_n, which is group of all permutations of set A, if A is the finite set $\{1,2\ldots n\}$. In the group theoretic solution representation, a vehicle trip is formulated by the cyclic factor in solution's disjoint cyclic structure. For example, a solution with the trips of two vehicles 1 and 2, *i.e.* 1 → 4 → 5 → 1 and 2 → 3 → 2 is represented as (1, 4, 5)(2, 3). Here, the first letter in the each tour represents the vehicle and subsequent vertices denote the customers served by it before returning back to the depot/hub. The tours with unit element represent non existent routes *i.e.* vehicle trip idleness. Each vehicle trip and customer service is assigned a different number. In mathematical terms, V and C denote the disjoint vehicle and customer sets with the condition that $|V|<|C|$, *i.e.* first all the vehicle trips are designated numbers and the successive numbers are allocated to the customer services. Further insights of this type of representation can be found in Crino [2].

3.2.2 *Schedule Generation*

The vehicle schedule is always to be generated as a measure for quality of solution. Once the schedule is generated the objective function value is obtained from equation 1. The Vehicle Loader/Scheduler Evaluation Heuristic [2] has been used to obtain the schedule.

3.2.3 *Neighborhood Generation*

The neighborhood generation is the most crucial stage in the implementation of any algorithm over the TDVRSP. Neighborhood generation scheme decides the search path adopted by the algorithm. This section first presents the two types of neighborhoods that are generated for the first four basic algorithms and later on the group theoretic neighborhoods corresponding to further four algorithms have been defined. These are:

Neighborhood Generation Schemes for first four algorithms: The two neighborhood generation, which is adopted in this section are - *within route neighborhood*;

and *between route neighborhoods*. The first one is based on *k*-opt [14] improvement for the Traveling Salesman Problem (TSP). The second type of neighborhood (between the routes) takes into account the exchanges of customers or a set of customer of two routes within a tier. Two types of *between the route neighborhoods* have been used and these are - (a) *String Relocation Method*: This procedure inserts a customer or a group of customers from one route to another route within the same tier. (b) *String Exchange Method*: This move exchanges two customers or two sets of customers between every two routes within the same tier.

Group Theoretic Neighborhoods for the other four algorithms: Group theoretic neighborhoods are a collection of moves specifically designed to suit the TDVRSP and corresponding group representation requirements. Four group theoretic neighborhoods, each defined for specific purposes have been utilized for this research, these are: (a) *Inter-orbital plane swap move neighborhood*: This move is similar to swap move with two customers from disjoint sets generated in *pre*Tabu Search phase. These moves are initially generated and stored, and later are used when called upon. (b) *Fill demand insert move neighborhood*: This move is invoked in order to get the solution with minimized demand shortfall. (c) *Inter-conjugacy class extraction move neighborhood*: This move specifically aims to generate neighborhood that takes care of the time definite delivery shortfall. These neighborhoods are generated sequentially in order to get a better and feasible solution to the problem. These neighborhood generation schemes are utilized by the following algorithms utilized in the portfolio analysis. The generation mechanism of the Group theoretic neighborhoods is more specifically detailed in [2].

3.3 Evaluating a Portfolio and Performance Measures

A Portfolio is evaluated on the basis of risk associated with it. The major aim to construct a portfolio is to gain computational advantage and quality edge while solving the problem at hand. The expected computational cost associated with a portfolio is the expected value of 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 the present case, let $\Re = \{\Re_1, \Re_2, \ldots, \Re_n\}$ be the set of random variables associated with n algorithms under consideration. $\Re$ is defined on a set of discrete outcomes, thus value of each $\Re_i \in \{1, 2, \ldots b\}$, where b denotes expected outcomes of different trials. Each outcome i is the number of function evaluations utilized to attain the prespecified performance level.

3.3.1 Mathematical & Statistical Evaluation

Let there be $A = \{A^1, A^2, \ldots, A^{|A|}\}$ algorithms embedded in a portfolio with P processors such that any algorithm $i \in A$ run over A_p^i processors. Hence, $P = A_p^1 + A_p^2 + \cdots + A_p^{A_p}$. As defined above, a random variable $\Re$ is associated with the portfolio. Let R be the combined probability distribution for all A_i'$s \in A$.

Further, for an algorithm A_i, it can be argued that $P(\Re_i = r)$ is defined as the probability that it requires r^{th} outcome to attain the prescribed objective performance [13]. The experiments can be seen as a sequence of Bernoulli trials, which instead of counting the number of successes in a fixed number of trials, count the number of trials until the first success (*i.e.* attaining specific performance level). If a failure be denoted by *0* and a success by *1*, then the sample space S of the experiments consist of the set of all binary strings with an arbitrary number of *0*s followed by a single 1. $S = \{0^{r-1}1 \mid r = 1,2,3\ldots\}$. Again, $\Re_i$ is the random variable such that the value assigned to the sample point $0^{r-1}1$ be r. In order to obtain the pmf (Probability Mass Function) of $\Re_i$, it is evident to that the event $P[\Re_i] = r$ is true *iff* there is a sequence of *r-1* failures followed by one success, as the case in proposed approach. If probability of each success is p, then the *pmf* of random variable $\Re_i$ is given by

$$p_{\Re_i}(r) = p(1-p)^r \qquad r = 1,2\ldots b \qquad (2)$$

The probability distribution function (pdf) of $\Re_i$, can thus be given by

$$F_{\Re_i}(t) = \sum_{r=1}^{t} p(1-p)^{r-1} = 1-(1-p)^t \;\; for \;\; t \geq 0 \qquad (3)$$

Let, first the case of two algorithms A_i and A_j, associated with random variables $\Re_i$ and $\Re_j$, running on different processors be analyzed. Let there be another random variable $\Re$, which defines the event that at least one algorithm reach the prescribed success level. Thus,

$$\Re = min\{\Re_i, \Re_j\} \qquad (4)$$

Lemma 4.1
The random variable $\Re$ is geometrically distributed given the two associated random variables $\Re_i$ and $\Re_j$ are also geometrically distributed and independent.

Proof: The probability $P(\Re > t)$ can be calculated provided the independence [13]; the independence of the algorithms is axiomatic as the processors considered in the paper have no interlinkage. Hence,

$$P\left(\Re > t\right) = P\left(\Re_i > t \,\&\, \Re_j > t\right) = P\left(\Re_i > t\right)P\left(\Re_j > t\right) \quad (5)$$

In terms of pdf,

$$P\left(\Re > t\right) = 1 - F_{\Re}\left(t\right) = 1 - \left[1 - F_{\Re_i}\left(t\right)\right]\left[1 - F_{\Re_j}\left(t\right)\right]$$

$$\Rightarrow F_{\Re}(t) = F_{\Re_i}\left(t\right) + F_{\Re_j}\left(t\right) - F_{\Re_i}\left(t\right)F_{\Re_j}\left(t\right) \quad (6)$$

From equation 2,

$$F_{\Re_i}\left(t\right) = \sum_{r=0}^{t} p(1-p)^t = 1 - p(1-p)^r \quad (7)$$

Clearly, as per equation 6,

$$\begin{aligned} F_{\Re}(t) &= 2\left[1-(1-p)^r\right] - \left[1 - 2(1-p)^r + (1-p)^{2(r)}\right] \\ &= 1-(1-p)^{2r} = 1 - \left[(1-p)^2\right]^r \end{aligned} \quad (8)$$

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

On similar grounds, let the already defined case pertaining to overall portfolio is investigated with b processors and $|A|$ algorithms. The algorithm running over any processor i is associated with a random variable $\Re_i \mid \forall i \in \{1, 2, \ldots, b\}$, which are geometrically distributed and are mutually independent. Let, now the random variable $\Re$ is associated with the portfolio-the following lemma holds.

Lemma 4.2

The random variable $\Re$ associated with the portfolio of b processors is geometrically distributed with parameter $1-(1-p)^b$.

Proof: Extending the case of Lemma 4.1,

$$F_{\Re}(t) = 1 - \prod_{i=1}^{b}\left[1 - F_{\Re_i}(t)\right] = 1 - \prod_{i=1}^{b}\left[(1-p)^t\right] \quad (9)$$

Thus, the random number $\Re$ is geometrically distributed with parameter $1-(1-p)^b$.

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

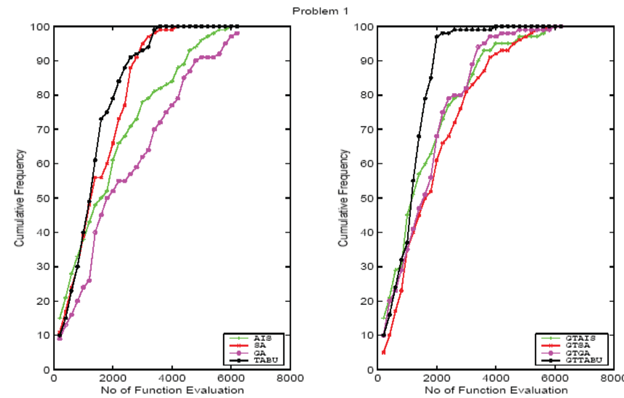

Figure 1: Results with 8 Algorithm-1 Processor System over Problem 1 (Frequency vs. Function Evaluations)

Lemma 4.3

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

$$F_{\Re}(t) < F_{\Re_i} \quad (10)$$

Proof: Let, first the following relation be considered for analysis

$$f = F_{\Re}(t) - F_{\Re_i} = \prod_{i=1}^{b}(1-p)^t - (1-p) \quad (11)$$

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

$$f = (1-p)^t\left[\prod_{i=1}^{b-1}(1-p)^t - 1\right] \quad (12)$$

Again, since $(1-p)^t < 0, \prod_{i=1}^{b-1}(1-p)^t$ is bound to be less than zero, thus making the second term of the above product as negative. In a portfolio more than one algorithm or processor is talked about, *i.e.* $b>1$, hence the abovementioned fact leads to the conclusion that in all cases a portfolio is expected to work faster than the cases with single algorithm; hence, establishing equation 10

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

$$\begin{aligned} E\left[\Re_i\right] &= \sum_{r=1}^{\infty} rp(1-p)^{r-1} = p\sum_{r=1}^{\infty}\frac{d}{dq}(q^r) \qquad q = (1-p) \\ &= \frac{p}{(1-q)^2} = \frac{1}{p} \end{aligned} \quad (13)$$

Similarly for the case of b algorithms,

$$E[\Re] = \frac{1}{1-(1-p)^b} \tag{14}$$

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

$$Var[\Re_i] = \frac{(1-p)}{p^2} \tag{15}$$

$$Var[\Re] = \frac{1-\left[1-(1-p)^b\right]}{\left[1-(1-p)^b\right]^2}$$

From the above expression, it is suggestive that if the probability of success and failure at any stage be constant, which can be logical in 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, the lemma follows.

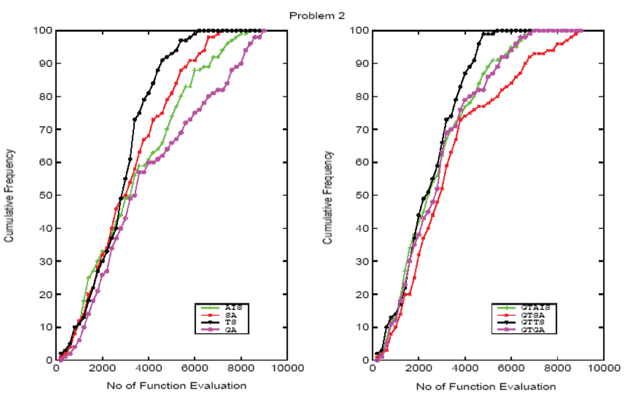

Figure 2: Results with 8 Algorithm-1 Processor System over Problem 2 (Frequency vs. Function Evaluations)

Lemma 4.4
Given the non varying mean ($E[\Re]=\mu$) and variance ($Var[\Re]=\sigma^2$), the pdf of the portfolio distribution has the following bounds

$$P\left(|\Re-\mu|\geq t\right) \leq \frac{\sigma^2}{t^2} \tag{17}$$

Proof: Intuitive from Chebshevs inequality [13].

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 the realistic outlook and evaluate the significance of various critically connected parameters.

3.3.2 Experimental Design

In order to initiate the experiments, first all the four algorithms and their corresponding group theoretic counterparts have been tested on the three problems under consideration on a single processor for 1000 runs and the results have been reported in Figures 1, 2 and 3.

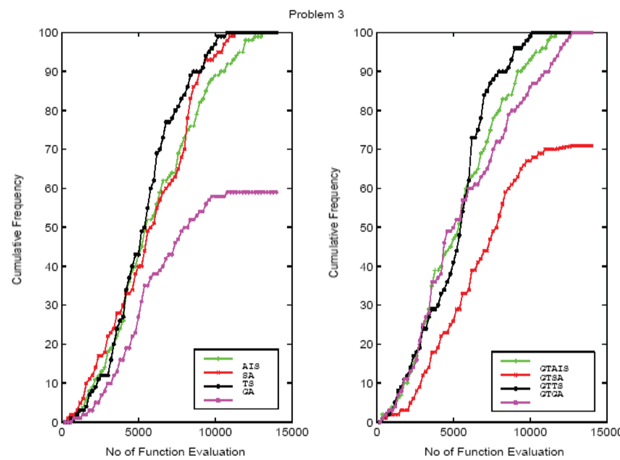

Figure 3: Results with 8 Algorithm-1 Processor System over Problem 3 (Frequency vs. Function Evaluations)

Table 2: Portfolio Design Scheme
(G2=GTGA; S1=SA; A1=AIS; A2=GTAIS; T1=TS; T2=GTTS)

No of algorithms	Case	2 Processors		4 Processors			
2	1	G2	A2	G2	A2	-	-
	2	S1	T1	S1	T1	-	-
	3	T1	T2	T1	T2	-	-
	4	A1	A2	A1	A2	-	-
4	1	-	-	A1	A2	T1	T2
	2	-	-	G2	S1	T1	T2
	3	-	-	S1	A2	G2	T2

For the above experiment, stopping criteria for each algorithm is set to be the performance level within 0.01% of best known objective value. Results are presented in terms of frequency of algorithm stops within a specific range of function evaluations. Here, in order to have better insight to the working of algorithm and to put 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 number of generations. Based on the figures 1, 2 and 3, it can be clearly gauged that most of the test algorithms work in a competitive manner, thus an obvious choice for portfolio. However, keeping in view the consistently inferior performance of simple GA and group theoretic SA over the rest, they have not been included in the further analysis and portfolio design has been confined to the cases of upto six algorithms. Thus, remaining experiments have been performed with various combinations of selected six algorithms viz. Group Theoretic GA (GTGA), Simple SA, Tabu search (TS), Group Theoretic TS (GTTS), simple AIS, Group Theoretic AIS (GTAIS).

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 and 4 processor systems with the designed portfolios of 2, 4 and 6 algorithms. The generalized scheme for designing portfolios has been presented in Table 2. With the settings mentioned in Table 2, each portfolio has been evaluated for different combination of selected algorithms and a '/' notation is used to denote such combinations. For example, the symbol *2/0* represent the case of 2 algorithms - 2 processor portfolio in which the first algorithm is run on both the processors and the second one is run over none. Each of such setting has been evaluated for *100* independent runs and the results have been analyzed on the average performance.

4. Computing with Portfolios: Results, Discussion and Statistical Insights

For the computational experiments, three standard problems of TDVRSP have been taken from literature, keeping in view their varying complexity and dimensions. Moving up in a consecutive manner, this section is parted in the a few subsections that sequentially present the theme and claims made in the paper.

4.1 Best Result

In order to have a realistic pictorial outlook of the theater and vehicle movement, best result obtained for Problem 1 has been plotted in a geometric coordinate plane shown in Figure 4. The presented result has been the one best obtained from the rigorous preliminary experimentations.

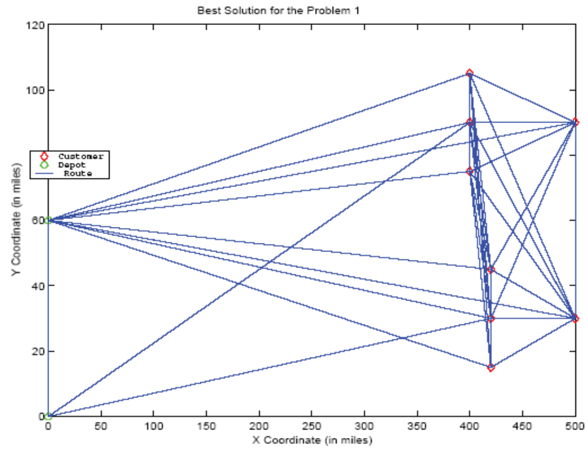

Figure 4: Route of the best solution for Problem 1

4.2 Experimental Runs

The experimentation has been divided in three parts on the basis of number of processors used. The experimental results have been presented as average number of generations for the portfolio to reach required quality level (0.01% of the best). The average has been calculated for 100 independent runs.

2 Processor System

For the two processor system, there are four selection of algorithms to be analyzed (Table 2). Each of the four cases is analyzed for the three possible combination of the algorithms viz. [2/0], [1/1], [0,2]. The portfolio has been tested on the three problems under consideration and the results obtained have been presented in Figure 5.

4 Processor System

The four processor system is characterized by the incorporation of 2 algorithms and 4 algorithm cases. For the 2 algorithm case, the possible combinations explored are [4/0], [3/1], [2,2], [1,3] [0,4]. With 4 algorithms case large number of combinations are possible. In order to be concise in exhaustive computational exercise, authors have selected 15 combinations for the study, however, the selection being intuitive. The results obtained by simulation runs for all these cases over the three problems are presented in Figures 6 and 7.

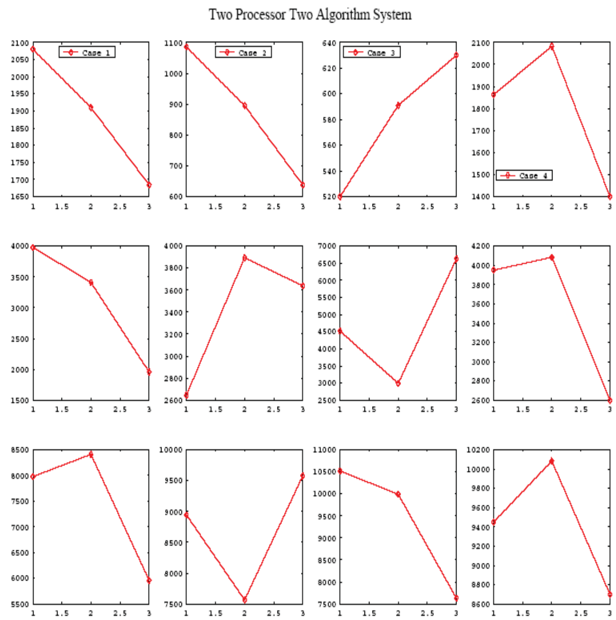

Figure 5: Results for Different Portfolio Cases Pertaining to Two Algorithm-Two Processor System

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.

4.2 Interpreting the Results

<u>*Best Portfolio*</u>: Having had the experimental results, the task to be accomplished is concerned to portfolio assessment. Generally processor availability for the parallel runs is limited in an organization; hence the selection strategy has to be executed separately for each of the two types of systems (2 processor and 4 processor) under consideration. The varying performances of different algorithms, and eventually the portfolios, pose enough challenge to select the best strategy among the instances explored in order to get the quality solution with minimum risk. Analytical Hierarchy Process (AHP) [15] has long been utilized as a tool to decision makers for selecting the best alternative from the given alternatives in such situations. AHP employs hierarchical pairwise comparison to induce the weights of alternatives thorough their pairwise comparison. This paper also addresses the selection of best portfolio problem from an AHP perspective.

Each type of portfolio is recognized as an alternative for the particular processor system to which the portfolio belongs to and the results over three different problems are considered as of different attributes. In this paper, the priority weights have been set as $1/2$ each. For the two processor system, a total of 12 (4 algorithm combinations $\times$ 3 portfolio combinations) alternatives have been evaluated. Similarly, for the 4 processor case with 65 alternatives. Figure 8 portrays corresponding final probability vector and ranks for the different alternatives discussed above.

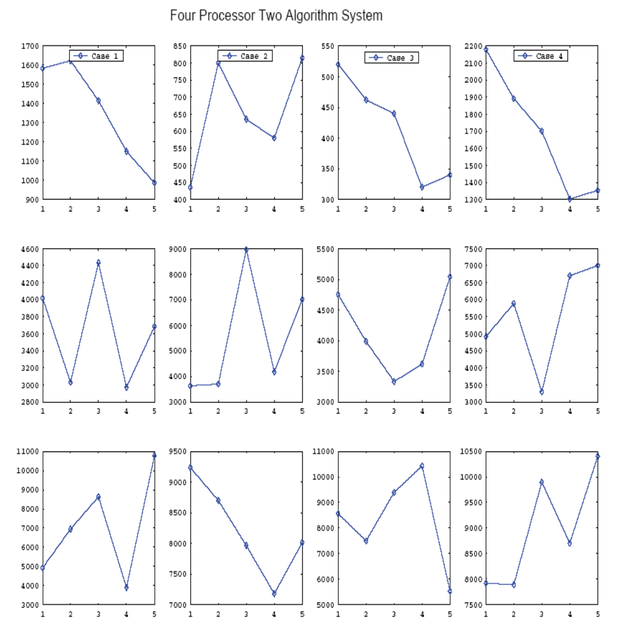

Figure 6: Results for Different Portfolio Cases Pertaining to Two Algorithm-Four Processor System.

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.

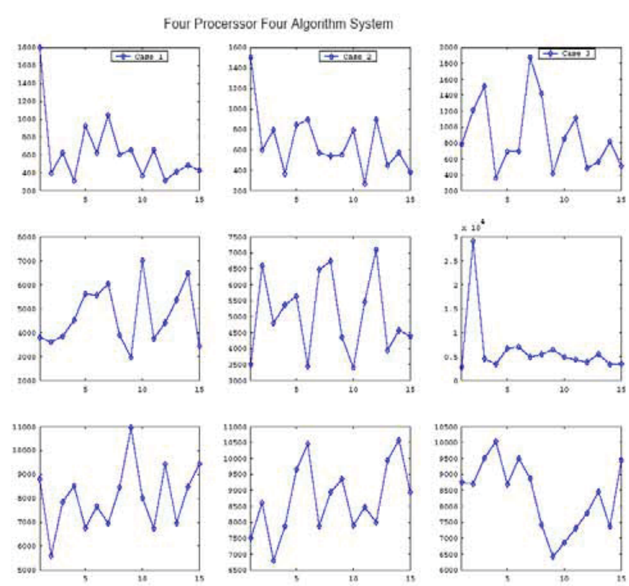

Figure 7: Results for Different Portfolio Cases Pertaining to Four Algorithm-Four Processor System

For the two processor case the portfolio with algorithms *AIS-GTAIS* characterized by (1/1) system is best. Similarly for 4 processor system, 7 cases have been found to be the best; however all correspond to algorithm set *SA-GTAIS-GTGA-GTTS*. In particular, AIS based strategies can be treated as computationally more viable than others under consideration. The above mentioned chosen alternatives can be recommended as the best suited strategies and as they are recognized as minimum risk portfolios that competitively perform well and takes care of the dimensional complexities in a better manner.

2 Processor – 2 Algorithm System			
Algorithm Selected	Portfolio	Priority Vector	Rank
G2 A2	2/0	0.496	3
	1/1	0.518	4
	0/2	0.769	12
S1 T1	2/0	0.628	7
	1/1	0.623	6
	0/2	0.658	9
T1 T2	2/0	0.666	10
	1/1	0.710	11
	0/2	0.633	8
A1 A2	2/0	0.468	2
	1/1	0.440	1
	0/2	0.603	5

4 Processor – 2 Algorithm System			
Algorithm Selected	Portfolio	Priority Vector	Rank
G2 A2	4/0	0.994	27
	3/1	0.993	24
	2/2	0.989	11
	1/3	1.003	55
	0/4	0.992	18
S1 T1	4/0	1.002	49
	3/1	0.996	31
	2/2	0.990	15
	1/3	1.000	40
	0/4	0.990	14
T1 T2	4/0	0.998	32
	3/1	1.002	46
	2/2	1.002	46
	1/3	1.004	50
	0/4	1.006	50
A1 A2	4/0	0.984	4
	3/1	0.984	3
	2/2	0.988	8
	1/3	0.985	4
	0/4	0.982	2

4 Processor – 4 Algorithm System			
Algorithm Selected	Portfolio	Priority Vector	Rank
A2 A2 T1 T2	4/0/0/0	0.988	4
	0/4/0/0	1.008	44
	0/0/4/0	1.000	30
	0/0/0/4	1.004	42
	3/1/0/0	0.993	12
	3/0/1/0	0.996	18
	3/0/0/1	0.991	8
	2/1/1/0	0.999	25
	2/0/1/1	0.998	21
	1/1/1/1	0.999	22
	1/2/1/0	1.001	27
	1/2/0/1	1.004	32
	1/0/2/1	1.002	28
	1/1/0/2	0.996	17
	0/3/1/0	1.003	29
G2 S1 T1 T2	4/0/0/0	0.992	9
	0/4/0/0	0.994	10
	0/0/4/0	0.997	16
	0/0/0/4	1.002	23
	3/1/0/0	0.991	7
	3/0/1/0	0.994	10
	3/0/0/1	0.995	10
	2/1/1/0	0.994	9
	2/0/1/1	0.998	12
	1/1/1/1	0.999	12
	1/2/1/0	1.004	19
	1/2/0/1	0.990	5
	1/0/2/1	1.000	13
	1/1/0/2	0.996	9
	0/3/1/0	1.002	14
S1 A2 G2 T2	4/0/0/0	1.000	11
	0/4/0/0	0.972	1
	0/0/4/0	0.987	2
	0/0/0/4	1.004	12
	3/1/0/0	0.992	4
	3/0/1/0	0.991	3
	3/0/0/1	0.985	1
	2/1/1/0	0.988	1
	2/0/1/1	1.001	5
	1/1/1/1	0.996	2
	1/2/1/0	0.993	1
	1/2/0/1	1.003	4
	1/0/2/1	0.996	1
	1/1/0/2	0.999	1
	0/3/1/0	1.001	1

Rank 1

Rank 2

Rank 3

Figure 8: Priority Vectors and Corresponding Ranking for Different Alternatives Related with the 2 Processor, 4 Processor

5. Conclusions and Future Research

This papers deals with a critical decision making combinatorial optimization problem that aims to generate feasible and optimized schedules for the vehicles moving within a theater. Portfolios are designed on the basis of parallel run of various algorithms without any interlinkage; absence of communication help algorithms take the advantage of inherent search capabilities and the parallel implementation ameliorates any disadvantage of 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. The conceptualization of portfolios to combinatorial optimization is believed to make a step to resolving the dilemma to choose the best search strategy among the nimiety of the search strategies, whether random or deterministic. The future work on introducing practical aspects like allowance in mobility of vehicles in different terrain, large scale emergencies, and involvement of crew members or human factors is going on.

References

[1]Position paper on the theater distribution management system (TDMS) initiative, *Headquarters, United States Air Force, Washington, DC*, (May 2000).

[2]J. R. Crino, A group theoretic tabu search methodology for solving theater distribution vehicle routing and scheduling problems, *Ph.D. Dissertation*, AFIT, (2002).

[3]J. A. CaxNo and J. T. Moore, Solving the Theater Distribution Vehicle Routing and Scheduling Problem using Group Theoretic Tabu Search, *Mathematical and Computer Modeling*, 39 (2004) 599-616.

[4]C. P. Gomes, B. Selman, Algorithm Portfolio Design: Theory vs. Practice, in: *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97), Morgan Kaufman Publishers, Providence, RI*, 1997, 190-197.

[5]B. Huberman, R. Lucose, T. Hogg, An economics approach to hard computational problem, *Science* 275 (1993) 51-54.

[6]Brandao J., Mercer A., A tabu search algorithm for the multi-trip vehicle routing and scheduling problem, *European Journal of Operational Research*, 100, 1997, 180-191.

[7]Glover, F., Laguna, M., and Marti, R., Principles of Tabu Search, In Approximation Algorithms and Metaheuristics, *Chapman & Hall/CRC*, 2005.

[8]De Castro, L. N. & Timmis, J. I., Artificial Immune Systems: A New Computational Intelligence Approach, *Springer-Verlag, London, September*, 2002, 357 p.

[9]M. Gen, and R. Cheng, Genetic Algorithm and Engineering Design, *Wiley*, 1997.

[10]Azencott R., Simulated Annealing: Parallelization Techniques, *New York: Wiley* 1992.

[11]Brown Kevin Leyton, Nudelman, Eugene, Galen, Andrew, McFadden, J., Shoham, Y., A Portfolio approach to algorithm selection, *International Joint Conference in Artificial Intelligence*, 2003.

[12]Rice, J. R., The algorithm selection problem, *Advances in Computers*, 15, 65–118, 1976.

[13]Trivedi, K. S., “Probability and Statistics with Reliability, Queuing, and Computer Science Applications,” *John Wiley \& Sons*, *Inc.*, 2003
[14]Lin, S., and Kernighan, B.W., An effective heuristic algorithm for the traveling salesman problem, *Operations Research* 21, 1973, 498-516.
[15]Pomerol, J. C., Barba-Romero, S., Multicriterion Decision in Management: Principles and Practice, *Kluwer Academic Publishers*, *Boston*, 2000.