

AN ANT BASED SIMULATION OPTIMIZATION FOR VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS

Mukul Tripathi
Glenn Kuriger
Hung-da Wan

Center for Advanced Manufacturing and Lean Systems
Department of Mechanical Engineering
University of Texas at San Antonio
San Antonio, TX 78249, USA

ABSTRACT

The Vehicle Routing Problem (VRP) is of considerable economic significance in logistic systems as it manages the distribution of goods to make an efficient transportation system. Considering a practical application, this paper solves a vehicle routing problem with stochastic demand (VRPSD) in which the customer demand has been modeled as a stochastic variable as opposed to conventional VRP. To deal with the additional computational complexity, this paper uses a simulation optimization approach to solve the VRPSD. To enhance the algorithm performance, a neighborhood-search embedded Adaptive Ant Algorithm (ns-AAA), an improved Ant Colony Optimization approach, is proposed. The performance of the proposed methodology is benchmarked against a set of test instances generated using Design of Experiment (DOE) techniques. The results verified the robustness of the proposed algorithm against Ant Colony Optimization and Genetic Algorithm, over which it always demonstrated better results, thereby proving its supremacy on the concerned problem.

1 INTRODUCTION

In a recent manufacturing scenario, firms need to deploy efficient methodologies to effectively schedule a fleet of delivery vehicles with an aim to minimize the total costs involved. Since design of optimal delivery and its allied services are the major cost promoters, the key issue for distribution companies involves determining best routes to a set of geographically scattered stops with respect to vehicle capacity, time windows (Lau, Sim, and Teo 2003) and drivers maximum working time (Bianchi et al. 2006) etc., while satisfying the customer demands. Therefore, the problem of efficient routing and scheduling of vehicles have become key concerns for the decision makers of related firms. In literature, such problems are classified as Vehicle Routing Problems (VRP) and arise as a wide range of practical decision making problems such as school bus routing, mail, newspaper delivery, fuel oil delivery and municipal waste collection (Tan, Lee, and Ou 2001). Moreover, structural similarity between VRP has been established with Traveling Salesman Problem (TSP), Bin Packing Problem (BPP) (Ralphs et al. 2000) and Generalized Assignment Problem (GAP) (Fisher and Jaikumar 1981).

One of the simplest instances of VRP can be framed as finding a feasible optimum solution for TSP which has been already shown as NP-complete in nature (Tan, Lee, and Ou 2001). Therefore, determining the optimal solution in VRP is even more complicated as it involves serving several customers using vehicles. To illustrate, Lau, Sim, and Teo (2003) explored a variant of VRP with time windows with a limited number of vehicles and proposed a Tabu search based approach marked by a holding list and a mechanism to enforce dense packing within a route. However, in many real-world applications, one or more parameters of VRP tend to be stochastic. These parameters can be the set of customers to be visited, customer demand, travel times or a combination of these. In general, stochastic variables of known probability distributions are utilized to model these parameters and the objective function thus formulated is expressed as the expected cost of the planned routes. These problems are referred to as the Stochastic Vehicle Routing Problem (SVRP). One common feature of SVRP is that they all have at least one deterministic VRP counterpart, which can be modified by mathematically modeling the deterministic variables to obtain the stochastic elements in the problem. Thus, with the inclusion of stochastic modeling the objective function formulated for SVRP tends to be computationally more expensive than conventional vehicle routing problems. Tillman (1969) solved the multiple terminal delivery problem with probabilistic demands where as a comprehensive modeling approach towards stochastic vehicle routing is presented by Stewart and Golden (1983). An intensive survey of models related to SVRP has been carried out by Gendreau, Laporte, and Seguin (1996). Cheong et al. (2006) considered a VRP with Sto-

chastic Demand (VRPSD) where customer demands were assumed to be stochastic and all other data known in priori. In this paper, we consider a VRPSD in which the customer demand has been modeled as a stochastic variable independently distributed with known distributions, such that its exact value is known only upon the arrival of the vehicles at the customer location. The model assumes that at some customer location, when the minimum vehicle load (load threshold) is attained, the vehicle must return to the depot for preventive restocking. However, if the residual load is greater than or equal to a load threshold, the vehicle proceeds to the next customer in usual fashion.

It has been empirically shown, that in the combinatorial optimization problems of the type defined above, optimality can be guaranteed only in relatively smaller problem instances through deterministic approaches. Tillman (1969) proposed a solution methodology for VRPSD based on Clarke and Wright's savings algorithm. However, to deal with the complexities inherent in combinatorial optimization problems of real dimensions, several stochastic search metaheuristics have been proposed. Tan, Lee, and Ou (2001) explored different hybridizations of artificial intelligence based techniques including Simulated Annealing, Tabu search and Genetic Algorithm for better performance in VRP with time window constraints. Gendreau, Laporte, and Seguin (1996) surveyed the development in the VRPSD with an emphasis on the insights gained and on the algorithms proposed. Cheong et al. (2006) presented a multi objective evolutionary algorithm that incorporates a specific heuristic for local exploitation and a route simulation method to evaluate the fitness of solution for VRPSD. Dong and Xiang (2006) proposed a novel hybrid algorithm that combines the Ant inspired meta-heuristics with local optimization heuristics.

This research follows a structured approach to determine optimal input parameter values for the VRPSD where optimality is measured by utilizing a simulation model to generate output performance variables. In general, the methodology combines simulation and an optimization algorithm to solve computationally complex problems and is widely known as simulation optimization in the literature. In existing VRPSD models, Ant Colony Optimization (ACO) (Dorigo and Gambardella 1997) has empirically shown superior results in resolution of computational complexity. Inspired by the successful applications of ACO, and with the increasing hunt for improvement in solution quality, this research proposes a new metaheuristic having its roots in traditional ant algorithms to develop a simulation optimization framework for optimizing the problem at hand. The proposed comprehensive modeling enhances the quality of the solution and decreases the computational time involved. To improve the search capability of the proposed algorithm, a neighborhood search based technique is utilized which attempts to enhance the quality of the solutions by using compound moves. In order to validate the efficacy and efficiency of the proposed technique simulations are performed over a set of benchmark instances. For the generation of the test bed, seven factors are selected, varied at two different levels, and their impact on the objective function values are investigated using Design of Experiments (DOE) (Taguchi 1987). The analysis also reveals the direct relationship between the objective function value and the parameters selected.

The rest of the paper has been organized as follows: Section 2 formulates the mathematical model of the VRPSD problem followed by section 3 where the proposed ant algorithm based simulation optimization solution methodology is detailed. Section 4 reports the experimental design and results showing the efficiency of the proposed algorithm. Finally, section 5 provides the conclusive remarks and states scope for future study.

2 MATHEMATICAL MODELING OF VRPSD AND RELATED CONSTRAINTS

Given an undirected 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, having a travel cost (distances) $C = \{c_{ij} : i, j \in V, i \neq j\}$ between the nodes (customers). The cost matrix "C" is symmetric and satisfies the triangular inequality. A fleet of homogeneous vehicles (HNV) each with vehicle capacity (V_c) has to deliver goods to the (N_c) customers according to their stochastic demands, minimizing the total expected distance traveled. We make the following assumptions before discussing the mathematical model of VRPSD. Customer's demands are stochastic variables θ_i , $i = \{0, 1, \dots, n\}$ which are independently distributed with known distributions. The actual demand of each customer is only known when the vehicle arrives at the customer location. It is also assumed that the demand θ_i does not exceed the vehicle capacity V_c , and follows a discrete distribution and probability mass function $\pi_{id} = \text{prob}(\theta_i = d)$, $d = 0, 1, 2, \dots, D \leq V_c$. All the vehicles have to visit the customers in the order specified by the priori tour. Let us assume that 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 the depot for restocking. Sometimes the choice of restocking is the better one, 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 vehicles when they do not have sufficient load to serve the preceding customer, thus it performs back and forth trips to the depot for completing the delivery at the customer. Let us define some of the variables that are crucial in formulating the constraints, these are as follows:

$X_{i,j,k}$: 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

$S_{i,k}$: 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

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

The objective function (OBF), i.e. expected distance covered by the vehicle, is computed as follows. Let \mathfrak{R} represent the priori tour (defined earlier) traversed by the HNV vehicles and the expected cost of the priori tour traversed by k^{th} vehicle is $f_{0k}(V_c)$. Therefore, the total expected cost for traversing the priori tour by all the HNV vehicles is denoted by variable OBF. 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, $f_{j,k}(\lambda) \forall \lambda \in L_{jk}$ satisfies equation (1).

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

where

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

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

The pseudo code shown below details the necessary computation for finding the objective function with the boundary condition $f_{nk}(\lambda) = c_{nk,0}, \lambda \in L_{nk}$. In equation (2) and (3) above, $f_{jk}^p(\lambda)$ is the expected cost corresponding to the choice of proceeding directly to the next customer. Whereas, $f_{jk}^r(\lambda)$ is the expected cost in case of preventive restocking by the k^{th} vehicle. As detailed by Yang, Mathur, and Ballou (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 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.

$$\text{Minimize } \{\mathbf{OBF} = \sum_{k=1}^{HNV} f_{ok}(V_c)\} \quad (4)$$

To calculate the OBF, equation (4), we use following formulation:

```

for (k = 1, 2, 3, ..., HNV) do
  for ( $\lambda = V_c, V_c - 1, \dots, 0$ ) do
     $f_{nk}(\lambda) = c_{nk,0}$ 
    for (j =  $n_k - 1, n_k - 2, n_k - 3, \dots, 1$ ) do
      //  $n_k$  is the last customer visited by the  $k^{th}$  vehicle
      compute  $f_{jk}^r$  using  $f_{(j+1)k}(\cdot)$  (using equation (3))
      for ( $\lambda = V_c, V_c - 1, \dots, 0$ ) do
        compute  $f_{jk}^p(\lambda)$  (using equation (2))
        compare  $f_{jk}^r(\lambda)$  and  $f_{jk}^p(\lambda)$  for finding the threshold  $LT_{jk}$ 
        compute  $f_{jk}(\lambda)$  using  $f_{(j+1)k}(\lambda)$  (using equation (1))
      end for
    end for
  end for
  compute  $f_{0k}(V_c)$ 
  OBF = OBF +  $f_{0k}(V_c)$ 
end for
return (OBF)

```

Where, OBF can be calculated as illustrated in the section of code shown above, subject to following constraints:

$$\sum_{k=0}^{HNV} X_{0,i,k} \leq HNV \quad \forall i \in V \tag{5}$$

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

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

Constraint equations (5) to (10) characterize the vehicle flows on the path. Equation (5) illustrates that the number of vehicles to service must not exceed available vehicles ready at the depot. Equations (6) and (7) represent the constraint that each vehicle flows from and back only to the supply node (denoted by 0).

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

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

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

$$S_{i,k} - \left[\sum_{j=0}^{|V|} Y_{i,j,k} - \sum_{j=0}^{|V|} Y_{j,i,k} \right] \geq 0 \quad \forall i \in V, k \in \{0,1,2\dots HNV\} \tag{11}$$

$$X_{i,j,k} V_c \geq Y_{i,j,k} \quad \forall i, j \subseteq V, k \in \{0,1,2\dots HNV\} \tag{12}$$

Equations (8) and (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.

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

$$X_{i,i,k} = 0, 1 \quad Y_{i,i,k} \geq 0 \tag{14}$$

$$\sum_{k=1}^{HNV} n_k = N_c \tag{15}$$

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. Finally, constraint equation (15) illustrates that the sum of customers traversed by *HNV* vehicles must be equal to *N_c*.

3 SOLUTION METHODOLOGY

The VRPSD is a combinatorial optimization problem and computationally complex in nature. Determining optimality in such problems requires all possible permutation of the sequences in which the vehicles have to be routed. Due to this complexity, finding optimal solutions (via deterministic approaches) even in smaller problem instances of real dimensions becomes almost impractical. In literature, ACO has empirically shown superior results in resolution of complexity for several NP Hard problems (Tripathi et al. 2009) and has found successful applications in many variants of vehicle routing problems (Dong and Xiang 2006). In this respect, this research follows a structured approach to determine optimal input parameter values for the VRPSD where optimality is measured by utilizing a simulation model to generate output performance variables. In general, the methodology combines simulation and the proposed algorithm to solve computationally complex VRPSD problem.

The optimization problem has been defined by the pair (O, CI), where O denotes the objective function to be evaluated and CI is the constrained input or the group of admissible decisions. If the value of O is provided in analytical form, then the problem reduces to a simple optimization problem. However, in various real-world cases, the value of O can only be evaluated as an output of a simulation experiment. In case of stochastic systems, only estimates for the objective function value can be given. Therefore, multiple replications are carried out to obtain an authentic result from a simulation experiment. However, just the calculation and replication of a simulation experiment does not solve the optimization problem. Therefore,

an optimization algorithm is integrated with the simulation model to provide the required near optimal results for the problem at hand. In simulation optimization we try to obtain the best input variable values (producing near optimal results) without explicitly evaluating each possibility. A generic simulation optimization framework for our problem is illustrated in Figure 1. The whole system framework is subdivided into two parts:

1. The Optimization Module
2. The Simulation Module

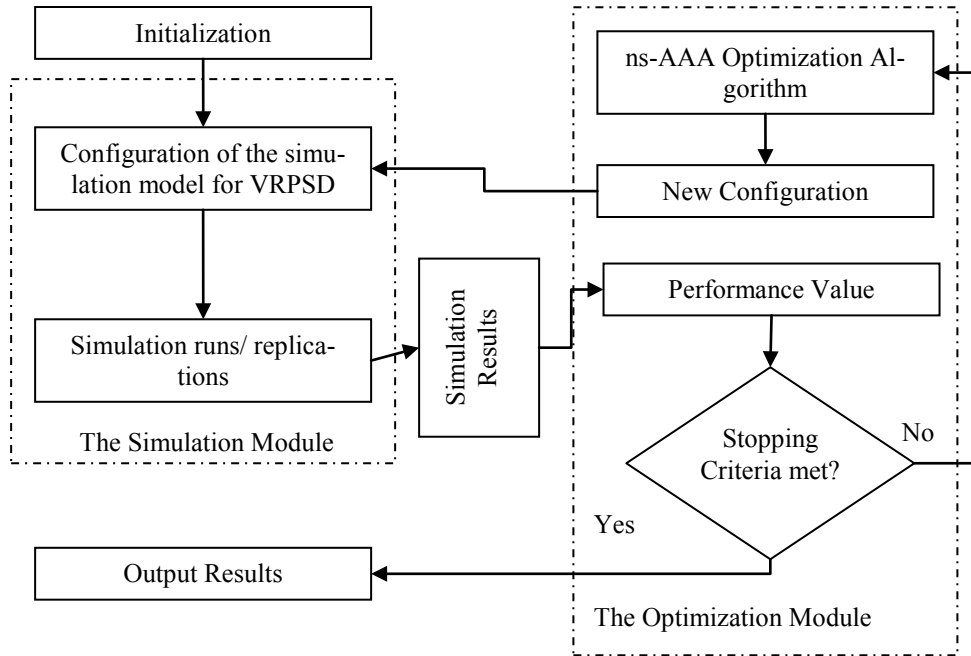


Figure 1: The Simulation Optimization Methodology

The system structure clearly reveals that the procedure of simulation and optimization are sequential in nature. The optimization algorithm is basically utilized to optimize the configuration parameters within the simulation model. It can also be seen that the simulation module is driven by the optimization algorithm which determines the new configurations with each iteration. The results of the simulation experiment provide the performance values to the optimization algorithm, which are utilized to keep the adaptive nature of the search strategy utilized. In this research, the authors have utilized a new search algorithm coined as ns-AAA. The simulation optimization embedded ns-AAA has been referred to as ns-AAA SO in the rest of the paper. The overview of the optimization strategy is discussed below.

3.1 Neighborhood Search Embedded Adaptive Ant Algorithm

In order to enhance the algorithm performance, we introduce a neighborhood search embedded Adaptive Ant Algorithm as an improvement to the existing Ant Colony Optimization (Dorigo and Gambardella 1997). This algorithm provides a resourceful metaheuristic approach having its roots in traditional ant algorithms. The proposed ns-AAA is constructed so as to deal with the problems of real dimensions such that the algorithm orients the search progressively in the directions favoring the global optimal solution. A fundamental aspect in the performance of this search strategy followed by ants is to identify an effective neighborhood for defining moves from one solution to another. The basic concept in this search metaheuristic is 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 and compound moves are usually obtained by combination of these moves. Osman (1993) uses a combination of insertion moves and exchange moves, vertex shifts from one route to another and exchanges vertices between routes based on the 2-opt process. Based upon this, the distinguishing features of the ns-AAA are detailed below:

1. A complete iteration is defined by two tours: the first tour (social tour) corresponds to the probabilistic movement of ants where as the second tour (neighborhood tour) is associated with the neighborhood generation.
2. Two pheromone update rules are utilized. A local pheromone update rule just after the social tour completion and a global pheromone update rule at the completion of both the tours.

- Each ant generates a complete tour by choosing the transition to the next city according to a probabilistic state transition rule, called the *adaptive-pseudo random proportional rule*. Throughout the run of the algorithm this rule acts as an adaptive and dynamic balance between the probabilistic exploration and the exploitation.

The progress of the modified algorithm proposed for the optimization problem is as follows:

3.1.1 The Social Tour

The algorithm is initiated by placing the ants randomly over the feasible cities. Here the number of ants is set equal to the number of cities. The number of cities is made equal to $N_c + HNV$. To force the ant to make a legal tour, transition to already visited cities are disallowed by forbidding the ant to move to those cities within the same tour. This constraint is enforced by associating a data structure called tabu list with each ant. Social tour starts when transitions are made to the unvisited feasible cities ($J_y(r)$) using a state transition rule by allowing the ants to move freely, where y represents the current iteration and r denotes the current city. The set ($J_y(r)$) is specific for each ant, and the cities that are in the tabu list are excluded from it. In this effect, ants are survived till their tour completion by avoiding the possibility of entrapment into any infeasible path. Now for the construction of the tour, the *adaptive-pseudo-random proportional rule* is utilized. This state transition rule provides an adaptive balance between the exploration and exploitation criteria during the run time and is depicted below,

$$s = \begin{cases} \arg \max_{u \in J_y(r)} \{ [\tau_y(r,u)] \cdot [\eta(r,u)]^\beta \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (16)$$

$$p_y(r,v) = \begin{cases} \frac{[\tau_y(r,v)] \cdot [\eta(r,v)]^\beta}{\sum_{u \in J_y(r)} [\tau_y(r,u)] \cdot [\eta(r,u)]^\beta} & \text{if } v \in J_y(r) \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

By equation (16), y is the iteration number, r is the city on which the ant is currently positioned and it chooses to move to a feasible city s . Here, q is the random real number uniformly distributed over the range $[0, 1]$. By equation (16) the parameter q_0 controls the exploration and exploitation criteria with the condition that $q \leq q_0$ for exploitation and $q > q_0$ for exploration. The term $\tau_y(r,u)$ is the long term information obtained from the pheromone level on the path between the cities r and u for the y^{th} iteration and the term $\eta[r,u]^\beta$ is the short term or myopic information obtained from the visibility matrix constructed using the information based on the problem. Further, the parameter β determines relative importance between the two factors viz, visibility and pheromone information. The adaptive-pseudo random proportional rule forces the ant to move only on the best available path to meet the exploitation criteria. If it fails i.e., $q > q_0$ the ant switches over to the exploration criterion given by condition S in equation (16) in which any ant stationed at city r , probabilistically selects city v for transition, with $p_y(r,v)$ transition probability. The condition $v \in J_y(r)$ guarantees that only the cities within the given set $J_y(r)$ get selected. Besides, all the infeasible cities have been assigned zero probability of selection. In equation (16) the condition $q \leq q_0$ forms a balance between exploitation and biased exploration. To avoid entrapment in the local optima, greater scope for exploration is preferable in the first few stages of the algorithm progress. Towards the final iterations of the algorithm progress, large exploitation is preferable. Due to the fact that the proposed metaheuristic is an intrinsically dynamic and adaptive search technique, an adaptive parameter handling strategy is suggested (Eiben, Hinterding, and Michalewicz 1999). The parameter q_0 is adaptively handled by the algorithm during the course of its progress according to following equation

$$q_0 = \begin{cases} 0.1 & \text{if } \frac{\log(y)}{\log(y_max)} < 0.1 \\ 0.9 & \text{if } \frac{\log(y)}{\log(y_max)} > 0.9 \\ \frac{\log(y)}{\log(y_max)} & \text{otherwise} \end{cases} \quad (18)$$

where y is the current iteration of the algorithm and (y_max) is the maximum number of iterations. Equation (18) states that

the parameter q_0 is allowed neither to fall below a minimum limit nor to surpass a maximum limit in order to provide limited opportunity for exploitation in the initial stages as well as some exploration in the final stages. Each such tour corresponds to the social tour by the ants.

3.1.2 The Neighborhood Tour

After the social tour completion, a set of ants communicate important information regarding neighborhood generation to the existing ants in the colony using three neighborhood generation strategies. For this purpose we select $3w$ number of ants randomly from the current tours and group them in three different sets: Multiple Exchange Set, Multiple Insertion Set and Multiple Reversal Set. w ants are placed in each of these sets. Each ant within a set is allowed to find an enhanced solution to the problem by utilizing a sequential ordering of the neighborhoods by using Multiple Exchange Scheme, Multiple Insertion Scheme and Multiple Reversal Scheme (Rego 2001) as shown in Figure 2.

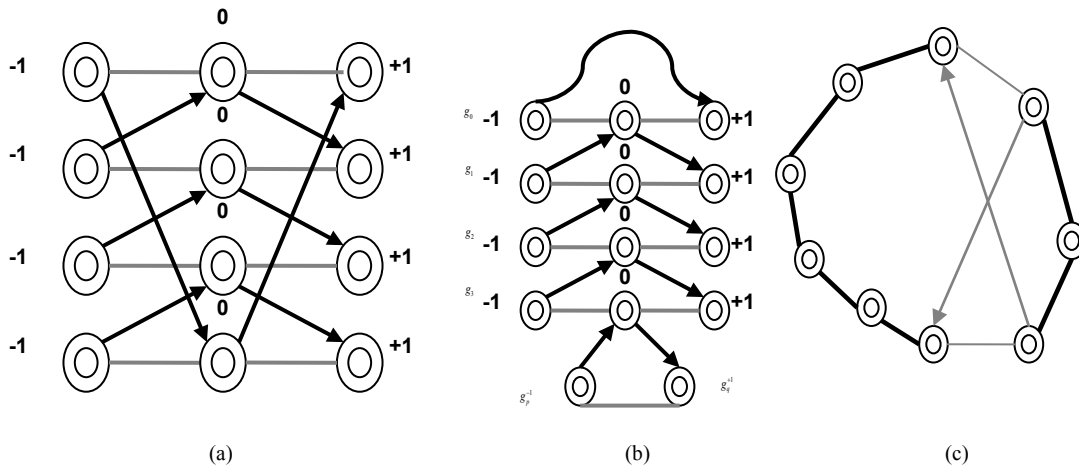


Figure 2: Three neighborhood generation techniques. (a) Multiple Exchanging Scheme, (b) Multiple Insertion Scheme, and (c) Multiple Reversal Scheme

In Multiple Exchange Scheme, several triplets of the node are taken from the ant’s solution string formulated as cyclic TSP and then multiple exchanging of the nodes (including customer and dummy nodes) are done by following the procedure: Let there be 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 is defined by:

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

Here T represents the set containing ‘ a ’ triplets. The process of multiple exchange is done by transforming T into T' as shown,

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

The transformation of T , i.e., $T \rightarrow T'$, shows a replacement of the central node of every triplet. Figure 2(a) schematically illustrates the multiple exchanging schemes detailed above.

In Multiple Insertion scheme, set (g_p^{-1}, g_q^0) is chosen excluding the nodes that are present in set T . Then this set is replaced by another set in T . By performing this move triplet T is transformed into T'' , where T'' is defined as:

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

This type of move has been illustrated in Figure 2(b).

In Multiple Reversal Scheme, a type of neighborhood generation scheme, a part of the solution string is taken out and it is again added to the solution string by reversing the node order. The illustration of this scheme is given in Figure 2(c).

These new paths generated correspond to the neighborhood tours. For each $3w$ ants selected for these neighborhood

tours, the fitness value of each ant's new tour (the tour length for the corresponding TSP model of the problem or the objective function value of the solution generated) is compared with the original fitness value of the ant's tour. If the new fitness value is found better, the corresponding ant's tour is replaced by the new tour generated by it. This information pertaining to neighborhood tours is stored on the paths in usual fashion using the global pheromone update rules.

3.1.3 The Pheromone Update Rules

As discussed earlier, we employ two pheromone update rules while the ants construct the solution paths. These are the local pheromone update rule, applied while the ants construct the solution during the tour, and the global pheromone update rule, applied at the tour completion by ants.

The local pheromone update rule is applied immediately after the ants complete the social and just before the neighborhood tour starts. Mathematically we represent the local pheromone update rule as,

$$\tau_{y+1}(r, v) = (1 - \rho) \cdot \tau_y(r, v) + \rho \cdot \Delta\tau_l(r, v)$$

where $\tau_{y+1}(r, v)$ is the pheromone level after $(y+1)^{th}$ iteration. The parameter ρ is a real number such that $\rho \in (0, 1)$.

$$\Delta\tau_l(r, v) = \sum_{k=1}^{N_c + HNV} \Delta\tau^k(r, v)$$

$$\Delta\tau^k(r, v) = \tau_0$$

τ_0 is defined as a constant quantity having value equal to the initial pheromone level of the transition paths between the cities. However, it is noticeable that the local pheromone update rule is employed to reward only the paths generated by social tours.

The global pheromone update rule is applied after all ants have completed constructing a solution. The purpose of the global pheromone update rule is to encourage ants to search for paths in the vicinity of the best tour found so far and also to reward the paths generated by neighborhood tours. The global best ant is defined as the ant which corresponds to the best tour generated till the current iteration. It is worth mentioning here that global pheromone update is applied only after the neighborhood tours have been constructed by the ants designated for the purpose. These neighborhood tours are always taken into account when considering the criteria for selecting the global best ant. The pheromone dropped by the global best ant is defined as,

$$\tau_{y+1}(r, v) = (1 - \alpha) \tau_y(r, v) + \alpha \cdot \Delta\tau^{gb}(r, v)$$

where $\tau_{y+1}(r, v)$ pheromone level after $(y+1)^{th}$ iteration. α is a parameter governing pheromone decay such that $0 < \alpha < 1$ and

$$\Delta\tau^{gb}(r, v) = \frac{Q}{L_{gb}}$$

where L_{gb} determines the quality of the tour constructed by the globally best ant and Q is the model dependent parameter. By introducing the neighborhood tours in ACO, its search capability is drastically enhanced; the results confirm this statement. The minimum level of pheromone trial level is explicitly set over each path (τ_{min}) to make the proposed algorithm more robust against entrapment into the local optima and include faster convergence towards the global optimum. Minimum pheromone level (τ_{min}) is maintained at each edge by,

$$\tau_{y+1}(r, v) = \begin{cases} \tau_{min} & \text{if } \tau_{y+1}(r, v) < \tau_{min} \\ \tau_{y+1}(r, v) & \text{otherwise} \end{cases}$$

This ensures a finite possibility of exploration of each feasible path and global optimum is highly likely to be attained in extended runs.

3.1.4 Visibility of the ants

Ants construct a solution in an iterative manner by using their short visibility (myopic information or greedy heuristic) with visibility matrix of order $n_c \times n_c$. Here visibility $\eta[r, v]$ between any two nodes r and v is the inverse of distance. If its distance is less then its visibility is more,

$$\eta[r, v] = \frac{1}{(d_{rv})}$$

where d_{rv} represents the distance between the nodes. Now the algorithm ends here after completing one tour. Immediately the ants expire and fresh ants are randomly positioned at feasible nodes for next iteration. This course of action is continued until (y_max) cycles are completed or the algorithm meets some other user specified criteria.

4 EXPERIMENTAL DESIGN AND RESULTS

This section details the experimental design utilized and results obtained by the application of ns-AAA SO methodology on the VRPSD. In order to study the effect of the variation of the parameters on the objective function value, we conducted a series of experiments over a carefully generated test bed. In this research, the objective function involves a number of design parameters out of which seven were identified to have greater impact on the objective function value. The effect of variation of these factors on the objective function value is studied with the help of Design of Experiments (DOE) (Taguchi 1987) by varying them at two different levels. Various permutations of these factors result in 2^7 different experiments. In order to avoid such an oversized test bed, we utilized Taguchi’s L_8 orthogonal array (Phadke 1989), which ensures a balanced combination of the levels of the factors and their interactions. The resulting test bed consists of a fractional set of 8 experiments but optimally represents the entire test domain of experiments. The first eight columns of Table 1 represent the Taguchi’s L_8 orthogonal array where 1 denotes low level and 2 high level. These levels are detailed as below.

Table1: Experimental design and Results of ns-AAA SO

| Prob. No. | N_c | HNV | D_i | S_j | V_c | r | Th_j | Objective function values | | | | | | | |
|-----------|-------|-------|-------|-------|-------|-----|--------|---------------------------|-------|-------|-------|-------|---------|---------------|---------------|
| | | | | | | | | Run1 | Run2 | Run3 | Run4 | Run5 | Average | $D_{min}(\%)$ | $D_{max}(\%)$ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8581 | 8592 | 8613 | 8614 | 8599 | 8599.8 | 0.2186 | 0.1651 |
| 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 14111 | 14119 | 14133 | 14108 | 14114 | 14117 | 0.0637 | 0.1133 |
| 3 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 16924 | 16987 | 16889 | 16872 | 16935 | 16921.4 | 0.2919 | 0.3876 |
| 4 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 8224 | 8131 | 8235 | 8229 | 8244 | 8212.6 | 0.9935 | 0.3823 |
| 5 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 16247 | 16221 | 16265 | 16302 | 16289 | 16264.8 | 0.2692 | 0.2287 |
| 6 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 40462 | 40439 | 40415 | 40481 | 40479 | 40455.2 | 0.0993 | 0.0637 |
| 7 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 28998 | 28976 | 28988 | 28915 | 28981 | 28971.6 | 0.1953 | 0.0911 |
| 8 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 19319 | 19314 | 19328 | 19330 | 19301 | 19318.4 | 0.0901 | 0.0600 |

The factors considered in this study and the two levels at which they are varied are discussed below:

1. Number of Customers (N_c): This factor determines number of nodes in the corresponding TSP model stated above and is directly related to the instance size and complexity. It was varied at a low level and a high level by choosing the value of N_c as 150 and 300 respectively.
2. Number of Vehicles (HNV): The total number of vehicles considered in this study were kept at 10 (low level) and 20 (high level).
3. Average demand (D_j): In the VRPSD the customer demand is a stochastic variable determined according to two parameters; D_j and Spread (S_j). We vary D_i at a higher level of 60 and lower level of 40.
4. Spread (S_j): Apart from the mean values, the stochastic customer demands also depend upon the variance S_j . Varying demand from minimum to maximum with vehicle capacity influences the problem instances. Variation of S_j was at 5 for lower level and 15 for higher level.
5. Capacity of vehicle (V_c): Vehicle capacity corresponds to the net maximum load that a vehicle can bear. Lower level and higher level of capacity vary at 100 and 200 respectively.
6. Distance (r, θ): Determines the customer location with polar coordinates r and θ with the distance varying in terms of r , lower level being (40, 60) and higher level being (80, 100).
7. Threshold(Th_j): Given the a priori tour for each customer j there is a load threshold (Th_j) such that if the residual load after serving j is greater than or equal to h_j , the better option is move forward action. Variation of threshold being taken at lower level as 10 and higher level as 20.

4.1 Parameter Settings

The simulation optimization framework is coded in MATLAB 7.1 and run on a workstation with 2 GHz Pentium D processor and 1 GB of RAM. We initialized the algorithm with τ and η matrices as described in the previous section and the minimum pheromone count τ_{\min} connecting the nodes on each path is set equal to 0.05. Numerous preliminary experimental runs were carried out before reporting the tuned values of the parameters for the best results obtained. The value of ρ governing the pheromone evaporation rate is set to 0.55 and the parameter value of α is set equal to 0.45 and $\beta = 3$. As stated earlier, the value of q_0 changes adaptively during the course of the algorithm and varies between 0.1 to 0.9 as shown in equation 18. The model dependent parameter Q is set equal to 5000 for $N_c = 100$ and 10,000 for $N_c = 200$. In this research, maximum number of iterations was chosen as the termination criteria. In our case we stopped the algorithm after 2000 iterations. There were 25 replications being done for each iteration within the simulation optimization framework.

For evaluation of performance, keeping the same simulation optimization framework, we have compared the results obtained from ns-AAA with the results from traditional Ant Colony Optimization (Dorigo and Gambardella 1997) and Genetic Algorithm (GA) (Thangiah 1995). Initially, we chose similar parameter setting for ACO as were for ns-AAA. But for tuned performance, we obtained the value of $\beta = 2$ and $\alpha = 0.5$. Moreover, initial pheromone level on each path was set equal to $\tau_0 = 0.5$ and q_0 was taken to be 0.4. In case of GA, we utilized a population size of 50. A two point crossover operator with probability 0.65 and bitwise mutation operator with probability 0.1 gave the best results. The terminating criteria for each of these two algorithms was also set as the same as that for ns-AAA.

4.2 Performance Comparison

Using Taguchi's Orthogonal Arrays ($L_8(2^7)$) as shown in Table 1, eight problem instances are designed by varying 7 factors at two levels to obtain maximum insights into the working of the model by performing limited number of experiments. The Table also lists the average results (for five runs) obtained after application of ns-AAA SO over VRPSD with different initial random seeds. From the same Table, we observe the significantly lower percentage values of the minimum and maximum deviation from the average objective function value in each experimental run. Among all the eight experiments designed using orthogonal arrays, the maximum values of D_{\min} and D_{\max} were found to be 0.9935% and 0.3876% respectively, which clearly shows the consistency in results obtained and thus the efficacy of the proposed ns-AAA SO framework.

The characteristic and the search capability of ns-AAA SO have been compared with ACO and GA as shown in Table 2 for problem instance 1 of the test bed. The results clearly reveal the supremacy of the proposed metaheuristic over the other two approaches compared, where ns-AAA SO was seen to significantly outperform them. The success of ns-AAA SO lies in the fact that the proposed metaheuristic adapts itself to maintain an adequate balance between exploitation and exploration throughout the run of the algorithm. Further, to prevent the search from entrapment into local optima, the minimum quantity of pheromone on any edge, τ_{\min} , is always maintained. This ensured the finite possibility of exploration of each feasible path and thereby increases the probability of attaining the global optimum.

Table 2: Comparison of average results obtained from GA, ACO and ns-AAA SO

| Problem instances | GA | ACO | ns-AAA SO |
|-------------------|----------|----------|-----------|
| 1 | 10037.88 | 9426.03 | 8599.8 |
| 2 | 17143.68 | 15473.16 | 14117 |
| 3 | 20546.46 | 18546.95 | 16921.4 |
| 4 | 10284.57 | 8904.038 | 8212.6 |
| 5 | 19535.83 | 16670.86 | 16264.8 |
| 6 | 49568.67 | 44317.14 | 40455.2 |
| 7 | 35443.97 | 31367.32 | 28971.6 |
| 8 | 24768.38 | 20919.73 | 19318.4 |

Further to present detailed analysis and to estimate the convergence trends of the algorithm the best fitness values obtained from ns-AAA, ACO and GA were plotted against total number of fitness evaluations in Figure 3. The distinguishing feature of ns-AAA SO is that it adaptively allows for higher initial exploration and promotes exploitation at later stages. This

is evident in Figure 3 where ns-AAA SO goes down steeply in the starting phase of the algorithm (maximum exploration) and thereafter searches in the vicinity of the best results found so far (by increasing the exploitation criteria gradually) as shown by the relatively smooth trends in the algorithm behavior towards its end.

The relative importance of visibility over the trail weight is a critical factor governing the quality of results obtained by the ant algorithms. The effect of variations in the β on the objective function value has been plotted as shown in Figure 4. To start with, equal preference was given to visibility and pheromone information by considering $\beta = 1$. We observed that giving more preference to pheromone information produced better results. However, when β was increased to high values, the objective function value started decreasing. In this research, an optimum balance was found to exist around $\beta = 3$.

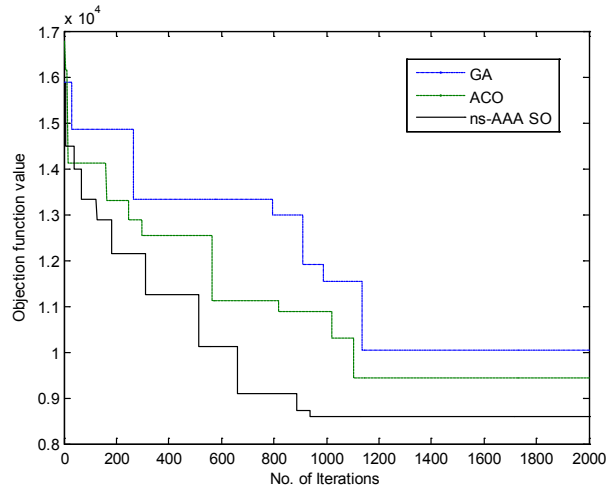


Figure 3: Convergence trends of ns-AAA SO, ACO and GA

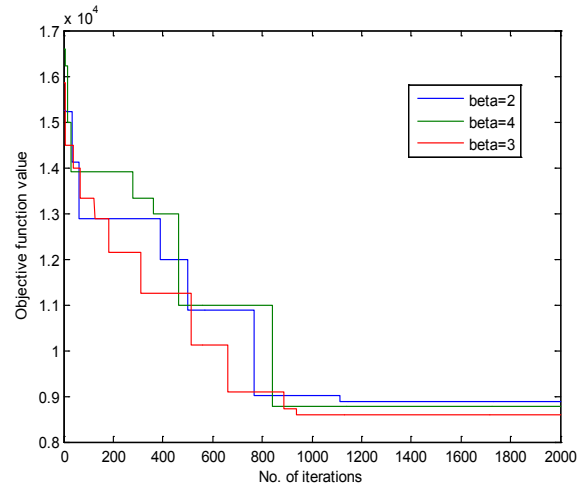


Figure 4: Effect of β (beta) values on ns-AAA SO performance

5 CONCLUSION

In this paper, a vehicle routing problem with stochastic demands (VRPSD) is undertaken for study, and a new algorithm entitled Neighborhood Search embedded Adaptive Ant Algorithm (ns-AAA) is utilized with simulation optimization. This research follows a structured approach to determine optimal input parameter values for the VRPSD where optimality is measured by utilizing a simulation model to generate output performance variables. In order to study the effect of the variation of the parameters on the objective function value, we conduct a series of experiments over a carefully generated test bed. The characteristic and the search capability of ns-AAA SO have been compared with ACO and GA. Furthermore, effects of variation of various parameters of ns-AAA SO have also been discussed. The results show that ns-AAA significantly outperforms the two algorithms compared, therefore, stating its efficacy. As a scope for future work, this research will be expanded to investigate the VRPSD with stochastic customer locations and other potential extensions using the ns-AAA SO methodology.

ACKNOWLEDGEMENTS

The work presented in this paper has been supported, in part, by the Air Force Research Laboratory, contract no. FA8650-08-C-6873.

REFERENCES

- Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. 2006. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms* 5(1):91-110.
- Cheong, C. Y., K. C. Tan, D. K. Liu, and J. X. Xu. 2006. A multiobjective evolutionary algorithm for solving vehicle routing problem with stochastic demand. *Proceedings of the 2006 IEEE World Congress on Evolutionary Computation*, July 16-21, 2006, Vancouver, BC, Canada, 5520-5526.

- Dong, L. W., and C. T. Xiang. 2006. Ant colony optimization for VRP and mail delivery problems. *IEEE International Conference on Industrial Informatics*, 1143-1148.
- Dorigo, M., and L. M. Gambardella. 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transaction on Evolutionary Computation* 1(1):53-66.
- Eiben, A. E., R. Hinterding, and Z. Michalewicz. 1999. Parameter control in evolutionary algorithms, *IEEE Transactions on evolutionary Computation* 3(2):124-141.
- Fisher, M. L., and R. Jaikumar. 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11:109-124.
- Gambardella, L. M., E. D. Taillard, and M. Dorigo. 1999. Ant colonies for the QAP. *Journal of the Operational Research Society* 50(2):167-176.
- Gendreau, M., G. Laporte, and R. Seguin. 1996. Stochastic vehicle routing. *European Journal of Operational Research* 88:3-12.
- Lau, H. C., M. Sim, and K. M. Teo. 2003. Vehicle Routing Problem with time windows and a limited number of vehicles. *European Journal of Operational Research* 148:559-569.
- Osman, I. H. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41:421-451.
- Phadke, M. S., 1989. Quality Engineering Using Robust Design. Prentice Hall, Englewood Cliffs, New Jersey.
- Ralphs, T. K., L. Kopman, W. R. Pulleyblank, and L. E. Trotter. 2000. On the Capacitated Vehicle Routing Problem. *Mathematical Programming Ser. B* 94:343-359.
- Rego, C. 2001. Node Ejection Chains for the Vehicle Routing Problems: Sequential and Parallel Algorithms. *Parallel Computing* 27(3):201-222
- Stewart, W., and B. Golden. 1983. Stochastic Vehicle Routing: A Comprehensive Approach. *European Journal of Operations Research* 14:371-385.
- Taguchi, G., 1987. *System of experimental design*, Vols. 1 and 2. White Plains, New York: UNIPUB/Kraus International Publications.
- Tan, K. C., L. H. Lee, and K. Ou. 2001. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. *Engineering Applications of Artificial Intelligence* 14: 825-837.
- Thangiah, S. R. 1995. Vehicle routing problems with time windows using genetic algorithms. In *Practical Handbook of Genetic Algorithms: New Frontiers*, ed. L. Chambers, Vol. II:253-277. Boca Raton, FL: CRC press.
- Tripathi, M., S. Agrawal, M. K. Pandey, R. Shankar, and M. K. Tiwari. 2009. Real world disassembly modeling and sequencing problem: optimization by algorithm of self guided ants (ASGA). *Robotics and Computer Integrated Manufacturing* 25(3):483-496.
- Tillman, F. 1969. The multiple terminal delivery problem with probabilistic demands. *Transportation Science* 3:192-204.
- Yang, W., K. Mathur, and R. H. Ballou. 2000. Stochastic vehicle routing problem with restocking. *Transportation Science* 34(1):99-112.

AUTHOR BIOGRAPHIES

MUKUL TRIPATHI is a Graduate Research Assistant at Flexible Manufacturing and Lean Systems Laboratory at University of Texas at San Antonio (UTSA). He received his Bachelors in Technology Degree from National Institute of Foundry and Forge Technology, India in 2008. His research interests include engineering management, workforce forecasting, artificial intelligence techniques and their variants, application of self-healing mechanism on manufacturing systems, modeling real world disassembly sequencing problems, multi-agent system design and application for multi-station assembly processes, product platform planning and optimization, and reliability optimization. His email is <mukul.tripathi@utsa.edu>.

GLENN KURIGER is a Research Assistant Professor at the Center for Advanced Manufacturing and Lean Systems (CAMLS) and the Department of Mechanical Engineering at the University of Texas at San Antonio (UTSA). He received his Ph.D. degree in Industrial Engineering from the University of Oklahoma. His current research interests include: simulation, operations research, simulation optimization, multi-criteria optimization, and lean manufacturing and lean concepts. His email address is <glenn.kuriger@utsa.edu>.

HUNG-DA WAN is an assistant professor in Mechanical Engineering and the co-director of Flexible Manufacturing and Lean Systems Laboratory at the University of Texas at San Antonio (UTSA). He received his PhD in Industrial and Systems Engineering from Virginia Tech. His research interests include lean manufacturing, web-based manufacturing systems, and computer integrated manufacturing. He is among the core faculty of the Center for Advanced Manufacturing and Lean Systems (CAMLS) at UTSA. His email is <hungda.wan@utsa.edu>.