MASTER

Self imposed time windows in vehicle routing problems

Li, Y.

Award date:
2011

Eindhoven, January 2011

Self Imposed Time Windows in Vehicle Routing Problems

By Yue Li
……………………

BSc Industrial Engineering
Student identity number 0666040

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
dr. T. van Woensel, TU/e, OPAC
dr.ir. N.P. Dellaert, TU/e, OPAC

TUE. School of Industrial Engineering.
Series Master Theses Operations Management and Logistics

Subject headings: vehicle routing problems with self-imposed time windows, tabu search metaheuristics, buffer allocation model

Abstract

This thesis considers a special version of the Vehicle Routing Problem (VRP) where a homogeneous fleet of vehicles with limited capacity may encounter stochastic disruptions on their way, which can be buffered by a series of time windows. In our problem – called VRP with self-imposed time windows (VRP-SITW), delivery time windows should be quoted by the route planners to their customers instead of being quoted by customers themselves, making our problem different from the traditional VRP with Time Windows (VRPTW). The vehicle must wait if it arrives before the time window opens and be penalized proportional to the tardiness if it arrives after the time window closes. In addition, returning to the depot after the shift time causes an overtime cost proportional to the lateness. The objective of this problem is to assign vehicles to feasible routes and make robust schedules that minimize the total costs, including travel cost, expected penalty cost for tardiness, and expected overtime cost. In this thesis, we describe an algorithm TSLPE to solve the problem, which develops a Tabu Search (TS) heuristic incorporated with a linear programming (LP) exact cost evaluation and an approximate cost evaluation. The basically designed TS is improved by adding a diversification procedure at a later stage. TSLPE is tested on a number of benchmark instances. Results show that the algorithm provides good quality solutions to our problem, while consuming reasonable computational efforts.

***Keywords*:** vehicle routing problem; vehicle scheduling; disruptions in travel times; tabu search; linear programming

Executive Summary

In recent years, numerous distribution centers have been constructed, aiming at reducing transportation costs. In order to improve the efficiency of distribution, the vehicle routing planners play a key role in the distribution system. The Vehicle Routing Problem (VRP) thus has received a lot of attention.

The VRP aims at designing a set of vehicle routes through several customer locations with minimum costs, in conditions that each route starts and ends at the depot, and each customer must be visited only once by one vehicle. Some side constraints may need to be satisfied, like the time windows during which it is allowed to service the customers, which constitutes an important variant of the VRP – the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is one of the central problems that have been extensively studied by researchers in distribution management and operations research.

The problem we consider in this thesis – called VRP with self-imposed time windows (VRP-SITW) – is a particular version of the VRP, where a homogeneous fleet of vehicles with limited capacity may encounter stochastic disruptions on their way, which can be buffered by a series of time windows. However, VRP-SITW is different from the well-studied VRPTW in two aspects. On the one hand, delivery time windows should be quoted by the route planners to their customers instead of being quoted by customers themselves (like in VRPTW). On the other hand, the VRP-SITW mixes the features of two important variants of the VRPTW – the VRP with Hard Time Windows (VRPHTW) stipulates that the service must start within the time window, and the vehicle can wait at no cost if it arrives too early, and the VRP with Soft Time Windows (VRPSTW) allows violation on both upper and lower bounds of the time windows with penalty proportional to the earliness or the tardiness. In VRP-SITW, the vehicle must wait at no cost if it arrives before the time window and be penalized proportional to the tardiness if it arrives after the time window closes. In addition, returning to the depot after the shift time causes an overtime cost proportional to the lateness. The objective of this problem is to assign vehicles to feasible routes and make robust schedules that minimize the total costs, including travel cost, expected penalty cost for tardiness, and expected overtime cost.

The VRP-SITW arises in a number of practical contexts where disruptions during travel might occur due to weather, human, or other unexpected factors. Thus distributors are motivated to impose time windows at customer locations to deal with these uncertainties. The time windows are often relaxed to allow for early or late arrivals at customer locations in practice. In applications where to increase customer satisfaction level is much more important (i.e., to respect the time windows as much as possible), the penalty for tardiness should be minimized. When costs on drivers are fixed during a shift, it is not necessary to associate penalty cost to the earliness. On the other hand, the distributors also want to minimize the travel cost and overtime cost for their self-interests. This problem is important because it can be used to model numerous problems in practice faced by distributors worldwide, e.g., newspaper delivery, postal delivery, and school bus routing. However, VRP-SITW has received little attention in the literature despite its importance in practice. A solution of the VRP-SITW includes three parts – the route planning, the service scheduling, and the expected total costs (evaluation). The objective of this thesis is to develop an algorithm to solve this problem, and evaluate its competitiveness through experiments.

VRPTW is a hard combinatorial optimization problem, and can rarely be solved to optimality for instances with large sizes within reasonable computation time. The same happens to VRP-SITW as well. We are therefore interested in using heuristics to solve this problem.

The algorithm we propose – called TSLPE – uses the ideas of Tabu Search (TS) heuristic, but incorporates problem-specific features. We construct a linear programming (LP) model to insert buffers (time windows) into the routes, in order to cope with the uncertainties and make a robust schedule. The objective function of the LP is used to exactly evaluate solution. However, since simply computing the objective function for all candidate solutions is computationally expensive, an approximation function has to be used to evaluate possible neighbors of a given solution, and then assist to select the best one as a new current solution. Consequently, the exact evaluation is only used on those selected elite solutions.

Specifically, TSLPE can be split into three parts, in order to obtain the three parts of the problem:

(1) The route planning method – the design of the TS framework

The TS is designed on six key factors. The initial solution is found through a Nearest Neighbor constructive heuristic. And the 2-opt and 2-opt* exchange operators generate the neighborhood. All the candidates generated are then evaluated using the approximation function, from which the one with the best value is selected as the 'current solution'. Afterwards, the exact function is computed to reevaluate the current solution. The tabu list records best moves from every iteration and has a variable length, and the memory structure contains both short-term and long-term memories. The aspiration criterion to allow the tabu status to be overridden requires that the exact value of a forbidden solution is better than the best value found so far. Finally, the termination criteria include the total number of iterations as well as the maximum number of non-improving iterations.

(2) The service scheduling method – the construction of the LP model

(3) The evaluation method – the construction of exact and approximate evaluation functions

The exact evaluation function summates the total travel time and the objective function of the LP model which computes the expected costs for tardiness and overtime. In addition, a self-adjusted penalty term is added to evaluate the infeasibility during the search. We consider two extreme situations and build two functions accordingly. The relationship between these functions and the real approximation function is examined using multiple regression technique.

TSLPE is tested on ten benchmark instances for VRP. The approximation evaluation and the neighborhood searching strategies for the TS design are determined through the experiments, and then some TS parameters are tuned on one randomly selected instance. Sensitivity analysis indicates that the effects of these parameters on solution quality are not large. Hence this new parameter setting is used for all datasets in later experiments. The TS procedure is improved by adding the long-term memory as a diversification method to pursue better solutions at a later stage. Comparisons are made between our results and the known optimal values in terms of the total distance. Results show that the improved algorithm provides good quality solutions to our problem, while consuming reasonable computational efforts, and is robust over penalty cost rate changes.

The contributions of this project include several aspects:
(1) The first to study on such a specifically defined VRP – VRP-SITW.
(2) The LP evaluation based on the buffer allocation model is once cooperated with TS in the machine scheduling problem, but never in the VRP. We propose a new method to incorporate the buffer allocation model into the process of TS for VRP-SITW, which provides satisfactory solutions.
(3) Presenting a practical and statistical way that is easily computed to estimate the objective function value constituted partly by the LP objective function which costs much computational effort during the mediate searching and proving its validation.
(4) One advantage of the proposed TSLPE lies in its flexibility. By raising the overtime cost rate, the shift duration becomes a hard constraint on the feasibility of the routes. By raising the penalty cost rate, hard time window cases may also be addressed. Other features can easily be handled too, such as assigning bounded vehicles, using multiple depots, allowing hierarchical routes, and so forth.

There are many perspectives that are worthy of receiving further investigation in future.
(1) First, as the parameters of the TS have been tested, the procedures of the other essential TS factors are also worth a try, e.g., more alternative strategies of generating an initial solution, more sophisticated neighborhood exploration to enrich the neighborhood, different memory structures, different aspiration criteria, more sophisticated diversification and intensification methods, post-optimization procedures, etc. The more successful implementations of TS often make use of better initial solutions and neighborhood structures, and a balance between intensification and diversification. More complicated strategies are likely to yield better solutions, but at the same time require additional computational effort. It should be remarked that the improvement of algorithms should strike a good balance between the quality of solutions and computational efficiency.
(2) Second, techniques that can speed up the search and improve the robustness and the quality of the solutions, such as multi-search meta-heuristics, or parallel computation, have been increasingly used in recent research, and may become a direction of future research.
(3) In addition, though lying out of the boundary of this project, the other parameters which are not directly related to the algorithm itself may find their optimized combination from a broader managerial view in order to further reduce the total costs of the whole business. For example, the effect of the shift duration or the penalty cost rate can be explored. Hence, many managerial insights can be taken into the tricky decision of the three cost rates and the shift duration in this way.

Acknowledgment

This master thesis is undertaken in the Department of Industrial Engineering & Innovation Science (subgroup of Operations, Planning, Accounting and Control), Eindhoven University of Technology. And it marks the end of my two-year master study in the program of Operations Management & Logistics in Eindhoven.

Time passed so quickly that suddenly it is the moment to say goodbye. It always makes me feel happy and sad when recalling the past two years, and thinking about things I've gained and missed. I feel lucky to have the opportunity that thousands of people admire and dream of – studying abroad. I would like to acknowledge with deep gratitude to Eindhoven University of Technology who gave me a two-year scholarship to support my study and life in the Netherlands, and teachers in Zhejiang University who helped me apply for the master program and the scholarship. I appreciate prof.dr.ir. J.C.Fransoo and prof.dr. A.G.de Kok who came in such a long way to Hangzhou, interviewed me and selected me. Without these people, I would never have the opportunity to continue my education and have such an unforgettable experience in the Netherlands.

I would like to express my gratitude to all those who helped me during the writing of this thesis. My special gratitude goes first to dr. T. van Woensel, my first supervisor who has walked me through all the stages of this thesis, for his constant encouragement and guidance. Without his consistent and illuminating instruction, this thesis would not have reached its present form. He not only instructed me on knowledge and research, but also let me learn lessons and understand principles. There is a plaque on the wall of his office, writing Chinese characters “doing is better than saying”, for which he always sets a good example. His rigorous and positive attitude impressed me a lot. I appreciate his encouragement and urge, his endless patience and tolerance on me. I would like to say sorry and thank you genuinely. I would also like to thank my second supervisor, dr.ir. N.P.Dellaert, for his professional guidance and valuable suggestions to help me improve my thesis. I would like to acknowledge the invaluable help of Ola Gabali, a kind big sister, who helped me with my research, and has been my constant consultant in my project. This thesis would not have been possible without the support and assistance that I received from her. I am grateful to all my friends, for their encouragement and great support. Thank Honggang for giving me lots of instructions on coding skills. Thank Junhan and Cheng whose love and support made me strong and brave. Thank Matthijs, my best Dutch friend for bringing me a lot of laugh in my life in the Netherlands. My special thanks should be given to my boyfriend, Yu, who offered me unselfish love and tolerance, took care of me, helped me through the tough time, and also gave me constructive suggestions on my thesis, in the past five months since we've been together. Thank you for making me happy, and making what I've been through worthy. Thank you for always being here for me.

Last but not least, my thanks would go to my families, for their endless love, selfless support and great confidence in me all through these years. Most particularly, I must thank my mother, who has been working so hard for 25 years to foster and educate me and my twin sister. Please accept my deepest apologies for giving you so much financial burden.

Finally, I wish to dedicate this thesis to my beloved grandpa, who passed away five months ago. I have been regretful for not being able to accompany you in the last time. May you rest in peace.

Yue Li
Eindhoven, January 2011

Table of Contents

Table of Notations

Table 1 the table of notations

Parameters	Explanation
γ	The self-adjusting coefficient of the penalty term for the infeasibility, its initial value is set to γ_0
Ψ_{ri}	Set of discrete disruption scenarios for customer i on route *r*. $i \in M_r$ $r \in R_r$
Δ_{rijk}	The delay in the starting time of serving customer *i* beyond the time window (e.g. the delay time that is punished) due to a disruption according to scenario *k* of the unique disruption at customer *j*. $r \in R_r$ $i, j \in M_r$ $i \geq j, i \neq 0$ $k \in \Psi_{rj}$
$\Delta_{\max}$	The largest observed absolute difference between the objective function values obtained at two successive iteration
A	Percentage of the total nodes that determines the size of neighborhood
B	The parameter used for updating γ
b_0,b_1,b_2,b_3	Coefficients in the regression model
C_o	Non-negative additional cost per unit time of overtime after the duration time
C_p	Non-negative penalty cost to pay the customer per unit time of delay at each customer after the time window
C_t	Non-negative traveling cost per unit time/distance of traveling
d_{rik}	*k*th largest disruption time in scenario set Ψ_{ri} at customer (or starting depot) *i* on route $r, i \in M_r$. $k \in \Psi_{ri}$ $r \in R_r$
$d_{\max}$	The longest disruption time over all disruption scenarios for all customers
D_{rijk}	The delay in the starting time of serving customer *i* or the delay at the starting depot due to a disruption according to scenario *k* of the unique disruption at customer *j*. $r \in R_r$ $i, j \in M_r$ $i \geq j$ $k \in \Psi_{rj}$
e	Prediction error in the regression model
F	The fixed common shift duration
$F(S)$	The evaluation function of current solution *S* (feasible or not)
$F'(S)$	The evaluation function of the improved algorithm
F_1^*	The best value of $F(S)$ so far for feasible solutions

F_2^*	The best value of $F(S)$ so far for infeasible solutions
$F_a(S)$	The approximated evaluation function of any solution (feasible or not)
$F_{a1}(S)$	The first approximation function
$F_{a2}(S)$	The second approximation function
F_{a1o}	The overtime cost for the first approximation function
F_{a2o}	The overtime cost for the second approximation function
F_{a2p}	The penalty cost for tardiness for the second approximation function
F_a^*	The best value of F_a so far for all candidate neighbors
$F_c(S)$	The cost function
$F_o(S)$	The objective function in the LP model
$F_u(v)$	The evaluation of the best solution found, which has the searching procedure fed by C_p=u but the evaluation function fed by C_p=v
f_v	The number of times move *v* has been moved, divided by the iteration number
g	Scaling factor
$g_{ri}(\bullet)$	Probability-mass function of disruption time at customer *i* on route r. g_{rik} is shorthand for $g_{ri}(d_{rik})$. $\sum_{k\in\Psi_{ri}} g_{rik} = 1 \;\; i \in M_r \;\; r \in R_r$
GAMA	The initial value of γ
H	The frequency of updating γ
I	Counter of the total iterations
I_FEASIBLE	Counter of the continuous iterations with feasible current solutions
I_INFEASIBLE	Counter of the continuous iterations with infeasible current solutions
INF	A large number standing for the concept of "infinity"
MAX_I	The maximum number of iterations the search is allowed to perform
MAX_NII	The maximum number of iterations the search is allowed to perform without improvement
MAX_L	Upper bound of the tabu tenure
MIN_L	Lower bound of the tabu tenure
M_r	A set of m_r customers and the depot, $M_r = \{0,1,2,\ldots,m_r\}$ in the route r. Vertex *0* denotes the depot. The original index of the customers are forgotten after the route is defined; instead, the customers are reassigned with new index from 1 to m_r according to their sequence. $r \in R_r$
NII	Counter of the continuous iterations without improvement

N_SIZE	The size of neighbors that each vertex is allowed to be moved with. The number is some percentage of the total number of customers.
p_{ri}	The probability that customer (or starting depot) *i* is the uniquely disrupted customer in route *r*, conditional on exactly one service being disrupted. $\sum_{i \in M_r} p_{ri} = 1 \quad r \in R_r$
Q	Capacity of a vehicle
q_{ri}	Demand of the (index-reordered) customer *i* on route *r*. $i \in M_r \quad r \in R_r$
R_r	A set of *R* routes (vehicles), $R_r = \{1,2, \dots, r, \dots, R\}$, where *R* is the total number of vehicles
S	The current solution (feasible or not)
*S**	The best feasible solution so far
TABU_LENGTH	The length of the tabu list
t_{ri}	Travel time (distance) between (index-reordered) customer (or starting depot) *i* and its next customer *i+1* (or returning depot) in sequence on route *r*. $i \in M_r \quad r \in R_r$
T_{ri}	The pre-scheduled time interval between customer (or starting depot) *i* and its next customer *i+1* in sequence on route *r*, after each pre-scheduled time point, the lateness of arrival can be *w* at maximum without a penalty charge $i \in M_r, i \neq m_r \quad r \in R_r$. Once all T_{ri} are determined, the schedules for all customers are fixed.
W	The length of the self imposed time window for all customers
Y_{rjk}	The delay of the tour *r* beyond the shift duration *F* due to the disruption of scenario *k* at *j*. $r \in R_r \quad j \in M_r \quad k \in \Psi_{rj}$

Table of Tables

Table of Figures

1 Introduction

In recent years, numerous distribution centers have been constructed, aiming at reducing transportation costs. In order to improve the efficiency of distribution, the vehicle routing planners play a key role in the distribution system. The Vehicle Routing Problem (VRP) thus has received a lot of attention. The VRP aims at designing a set of vehicle routes through several customer locations with minimum costs, in conditions that each route starts and ends at the depot, and each customer must be visited only once by one vehicle. Some side constraints may need to be satisfied, like the time windows during which it is allowed to service the customers, which constitutes an important variant of the VRP – the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW is one of the central problems that have been extensively studied by researchers in distribution management and operations research.

The problem we consider in this thesis – called VRP with self-imposed time windows (VRP-SITW) – is a particular version of the VRP, where a homogeneous fleet of vehicles with limited capacity may encounter stochastic disruptions on their way, which can be buffered by a series of time windows. However, VRP-SITW is different from the well-studied VRPTW in two aspects. On the one hand, delivery time windows should be quoted by the route planners to their customers instead of being quoted by customers themselves (like in VRPTW). On the other hand, the VRP-SITW mixes the features of two important variants of the VRPTW – the VRP with Hard Time Windows (VRPHTW) stipulates that the service must start within the time window, and the vehicle can wait at no cost if it arrives too early; and the VRP with Soft Time Windows (VRPSTW) allows violation on both upper and lower bounds of the time windows with penalty proportional to the earliness or the tardiness. In VRP-SITW, the vehicle must wait at no cost if it arrives before the time window and be penalized proportional to the tardiness if it arrives after the time window closes. In addition, returning to the depot after the shift time causes an overtime cost proportional to the lateness. The objective of this problem is to assign vehicles to feasible routes and make robust schedules that minimize the total costs, including travel cost, expected penalty cost for tardiness, and expected overtime cost.

The VRP-SITW arises in a number of practical contexts where disruptions during travel might occur due to weather, human, or other unexpected factors. Thus distributors are motivated to impose time windows at customer locations to deal with these uncertainties. The time windows are often relaxed to allow for early or late arrivals at customer locations in practice. In applications where to increase customer satisfaction level is much more important (i.e., to respect the time windows as much as possible), the penalty for tardiness should be minimized. When costs on drivers are fixed during a shift, it is not necessary to associate penalty cost to the earliness. On the other hand, the distributors also want to minimize the travel cost and overtime cost for their self-interests. This problem is important because it can be used to model numerous problems in practice faced by distributors worldwide, e.g., newspaper delivery, postal delivery, and school bus routing. However, VRP-SITW has received little attention in the literature despite its importance in practice. This objective of this thesis is to develop an algorithm to solve this problem, and evaluate its competitiveness through experiments.

VRPTW is a hard combinatorial optimization problem, and can rarely be solved to optimality for instances with large sizes within reasonable computation time. The same happens to VRP-SITW as well. We are therefore interested in using heuristics to solve this problem. The algorithm we propose – called TSLPE – uses the ideas of Tabu Search (TS) heuristic, but incorporates problem-specific features. We construct a linear programming (LP) model to insert buffers (time windows) into the routes, in order to cope with the uncertainties and make a robust schedule. The LP evaluation based on the buffer allocation model is once cooperated with TS in the machine scheduling problem (Finke,D.A., D.J. Medeiros, and M.T. Traband, 2007), but never in the VRPTW, at least at the author's knowledge. The objective function of the LP is used to exactly evaluate a solution. However, since simply computing the objective function for all candidate solutions is computationally expensive, an approximation function has to be used to evaluate possible neighbors of a given solution, and then assist to select the best one as a new current solution. Consequently, the exact evaluation is only used on those selected 'elite' solutions. Our algorithm is tested on a number of benchmark instances. The approximation evaluation and the neighborhood searching strategies for the design of TSLPE are determined through the experiments, some TS parameters are tuned, and the solution quality is checked. The TS procedure is improved by adding the long-term memory as a diversification method at a later stage. Results show that the improved algorithm provides good quality solutions to our problem, while consuming reasonable computational efforts.

The remainder of this report is arranged as follows. In chapter 2, a brief literature review of related topic is provided. In chapter 3, the problem of the project is defined and described. In chapter 4, 5 and 6, the algorithm followed by the model and evaluation is proposed. In chapter 7, the experiments and the results are presented and analyzed. And finally, the discussion and the conclusion are the subject of the last chapter.

2 Literature Review

In this chapter the literature review of related topics is presented. In a broader sense, the concept of VRP and approaches used in literature are introduced. And in a more specific sense, our main interests, namely VRPTW and TS, are studied as well.

2.1 VRP and VRPTW

In general, the VRP is described as the problem of designing optimal delivery or routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints (Laporte, 1992). However, there does not exist a single universally accepted definition of the VRP because of the diversity of constraints, such as vehicle capacity, route length, time windows, precedence relations between customers, etc (Laporte, 2007).

The VRP is NP-hard because it includes the Traveling Salesman Problem (TSP) as a special case when there is only one vehicle and the capacity is infinite. In early research (e.g. (Gendreau,M., A. Hertz, and G. Laporte, 1994)), solutions of the VRP are sometimes transformed into the TSP by replication of the depot. The most sophisticated exact algorithms for the VRP can only solve instances of up to about 100 customers, and with a variable success rate (R. Baldacci, N. Christofides, and A. Mingozzi, 2008). This explains to a large extent why most of the research effort has concentrated on heuristics. Another reason is the fact that heuristics tend to be considerably more flexible than exact algorithms and can be more readily adapted to deal with the diversity of variants arising in practice. For an introduction of different approaches refer to 2.2.

One important variant of the VRP is the Vehicle Routing Problem with Time Windows (VRPTW) where each customer has a given time window which needs to be respected. The VRPTW can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points. The routes must be designed in such a way that each point visited only once by exactly one vehicle within a given time interval, all routes start and end at the depot, and the total demands of all points on one particular route must not exceed the capacity of the vehicle (Braysy,O. and M. Gendreau, 2005a).

Many researchers, such as Braysy and Gendreau (2005b), consider that the VRPTW has multiple objectives in that the goal is to minimize not only the number of vehicles required, but also the total travel time or total travel distance incurred by the fleet of vehicles. A hierarchical objective function is defined such that the number of routes is first minimized and then for the same number of routes, the total traveled distance or time is minimized. However, this research will consider problems with a different objective function which will be discussed later in this report.

2.2 Solution Approaches

A wide variety of exact and approximate algorithms have been proposed for solutions of VRP/VRPTW. Exact algorithms, based on branch-and-bound techniques, can only solve relatively small problems (at most 100 customers), but a number of approximate algorithms have proved very satisfactory for large problems. The NP-hardness of the VRPTW requires heuristic solution strategies for most real-life instances.
Approximate algorithms include classical heuristics and metaheuristics.

(1) Classical Heuristics
The classical heuristics have improvement steps that always proceed from a solution to a better one in its neighborhood until no further gain is possible. They can be divided into constructive heuristics and improvement heuristics.

The most popular constructive heuristics is the Clarke and Wright savings algorithm (Clarke,G., and J.V. Wright, 1964), which puts every customer himself with the depot on one route at first, and then iteratively merges the two routes that cause the most saving unless the constraints are violated. Other important classes include, e.g., petal heuristics, the sweep algorithm (Gillett,B.E., and L.R. Miller, 1974), a heuristic based on a two-phase decomposition procedure (Fisher,M.L., and R. Jaikumar, 1981).

Improvement heuristics include intra-route and inter-route heuristics. Intra-route heuristics post-optimize each route separately by means of a TSP improvement heuristic; inter-route heuristics consist of moving vertices to different routes (Laporte,G., and F. Semet, 2002). The performance of classical improvement heuristics is often not so good. They are best used as building blocks within metaheuristics.

(2) Metaheuristics
In practice, metaheuristics is the core of many recent developments of solution methodologies for VRPTW, which allows the consideration of non-improving and even infeasible intermediate solutions, so that the exploration of the solution space beyond the first local minimum encountered is possible. All metaheuristics embed procedures borrowed from classical construction and improvement heuristics. Although it requires more computation time than classical heuristics, but given the vast improvements in solution quality, the extra computational effort is well justified (Gendreau,M., A. Hertz, and G. Laporte, 1994). Their solutions may be better than local optimum and sometimes even equal to the global optimal solution .

Metaheuristics can be classified into three categories: local search, population search, and learning mechanisms.

Tabu Search (TS) is one of the most successful local search methods to address the VRP in recent years, and is of particular interest to us. An introduction of TS is in 2.3.

2.3 Tabu Search

Tabu search (TS) is a local search metaheuristic introduced by Glover (1986). TS explores the solution space by moving at each iteration from a solution to the best solution in a subset of its neighborhood iteratively until a termination criterion is satisfied. The current solution may deteriorate from one iteration to the next in order to avoid local minima. New, poorer solutions are accepted only to avoid paths already investigated. This insures new regions of a problem's solution space will be investigated with the goal of avoiding local minima and ultimately finding the desired solution. To avoid cycling, solutions possessing some attributes of recently explored solutions are temporarily declared tabu or forbidden. The tabu status can be overridden if certain conditions are met. Various intensification and diversification techniques are often employed to guide the search process.

There are four basic elements in TS:

(1) **Initial solution**: typically created with some cheapest insertion heuristic. The most common is Solomon's I1 insertion heuristic (Solomon, 1987).

(2) **Neighborhood generation method**: some exchange operators are widely used to improve the solutions, e.g. 2-opt, Or-opt, 2-opt*, relocate, exchange, and CROSS-, GENI-, and λ-exchanges. To reduce the complexity of the search, some authors propose special strategies for limiting the neighborhood. Another frequently used strategy to speed up the search is to implement the proposed algorithm in parallel on several processors. To cross the barriers of the search space, created by time window constraints, some authors allow infeasibilities during the search. The violations of constraints (load, duration, time windows) are penalized in the cost function, and the parameter values regarding each type of violation are adjusted dynamically.

(3) **Stopping conditions**: total number of iterations and number of iterations without an improvement in the objective function value.

(4) **Tabu list**: include a simple single list strategy using a short-term list (recency-based memory) with a long-term list (frequency-based memory) for intensification and diversification, and using flexible memory structures where the list size changes over time. To save time and memory, it is customary to store not the tabu solutions themselves, but one of their attributes (Gendreau,M., G.Laporte, R.Seguin, 1996). Frequency can be integrated with recency to provide a composite memory structure.

Several tabu search implementations have been highly successful, starting with the algorithm of Taillard (1993) and its enhancement consisting of the use of an adaptive memory (Rochat,Y., and E.D. Taillard, 1995). These two algorithms have yielded some of the best known solutions on standard test instances. Other successful tabu search implementations include the Unified Tabu Search Algorithm (UTSA) which is highly flexible and applies to a wide variety of routing problems (Cordeau, J.-F., G. Laporte, A. Mercier, 2001), and Taburoute (Gendreau,M., A. Hertz, and G. Laporte, 1994) often produces the best known solutions, and also very flexible. Taillard et al. (1997) provide several approximations during the TS procedure, from which good exchanges are kept in the memory for further consideration, and then the best one according to the exact evaluation of those solutions are selected as the new current solution. Our current study incorporates many of the design features of Gendreau et al. (1994) and Taillard et al. (1997). A general and basic TS framework given by James et al. (2009) is shown in Figure 1.

```
Loop while (num_failures < max_failures)
        If is tabu but meets all aspiration criteria or is not tabu and best cost so far
                store best exchange that meets all conditions
        End If
        update tabu list
        make exchange on working solution
        If strictly improving
                update best solution
        Else
                increment num_failures
        End If
End Loop
```

Figure 1 General Tabu Search Framework

2.4 Buffer Allocation Model

One should notice that most successful methods combine several methodologies. Except TS, the method we propose is also based on the buffer allocation model. The buffer allocation problem is well-defined in the scheduling literature in the context of project and machine scheduling. It considers pursuing a stable and robust schedule with a common deadline for machines or projects in uncertain, disruptive environments. Buffers are inserted to deal with the uncertainties. Detailed descriptions about this problem and model are well documented in literatures, e.g., Herroelen and Leus (2004), Leus and Herroelen (2005), Leus and Herroelen (2007), and Ballestin and Leus (2008). The buffer allocation problem shares some common features with our problem. In the context of a machine scheduling problem, the jobs and the single machine can be compared to the customers and the vehicle on one route of a VRPTW. Therefore, a mathematical programming model is built in light of the buffer allocation model for our problem. In particular, a Linear Programming (LP) model has to be solved in order to evaluate the objective function of the problem.

3 Problem Analysis

In this chapter, the statement of the problem in our context is provided with notations, assumptions, the objective function and the relative constraints. At last, methodologies of the specific implementation are proposed.

3.1 Problem Statement

3.1.1 Notations and Definitions

A set of n customers $N=\{1,\ldots,n\}$ is to be assigned to several routes. Vertex 0 represents the depot, and every route must originate from the depot and end at the depot. A fleet of K identical vehicles with a fixed capacity Q and a fixed shift duration F are available to serve the customers and each customer i has a demand q_i (not more than Q). In this problem, the number of vehicles to be assigned is a fixed value rather than a decision variable, which is motivated by the applications where the routes followed by drivers have to remain almost the same from day to day. A traveling time t_{ij} which may include service time at customer i is associated to each arc between customer i and j. A self-imposed time window with length w is quoted to each customer. For each customer i or the depot discrete disruptions with length d_{ik} happen with a distribution given a scenario set Ψ_i and probability-mass function $g_i(\bullet)$.If the vehicle arrives too early, it must wait up to the lower bound to begin its service. If the vehicle is too late, a penalty for lateness is incurred. There are three fixed cost rates in the problem. A cost of C_t in practice relates to the unit cost of travel on the routes. A cost of C_p is associated to punish every unit time of tardiness later than the time window at each customer. A cost of C_o is associated to punish every unit time of delay over the shift duration finally after the tour is finished, which practically could refer to the overtime pay for the drivers after their shifts. All the above parameters are fixed and predefined. Their combination of values might be optimized in practice but this issue is out of the scope of this project. Decisions have to be made on both assigning customers to a set of routes, and scheduling a set of time intervals T_{ij} to each arc between successive vertices i and j so that the schedule of each vehicle can be determined. Other parameters are also necessary to be defined for modeling purpose. A comprehensive definition of all parameters is given in Table 1.

3.1.2 Assumptions

We make several assumptions:

(1) The travel time between two vertices is undirected and is proportional to travel distances. Furthermore, the triangle inequality is satisfied for the travel times.

(2) The service time of each customer is identical regardless the demand and thus can be ignored, or seen as included in the travel time. In the modeling, it is set to 0.

(3) All vehicles are identical. All the vehicles will be assigned.

(4) On every route, only one service time disruption can happen at a time. This assumption implicitly ignores the interactive effects between disruptions and provides the model with less complexity and is consistent with the assumption in the buffer allocation model. In practice, this model is useful when the disruption is sparse and spread over time, so that the disruptions are independent and can be considered separately.

(5) The lower bound of the time window constitutes an earliest starting time for the service operations. Arrival before the scheduled window is permitted, and the vehicles wait at no cost until service becomes possible. The arrival after the window or after the shift duration is also permitted, but with a cost proportional to the tardiness.

3.1.3 Objective and Constraints

The objective is to find a set of closed routes, such that the total weighted average cost is minimized, and the vehicles are not overloaded. The cost is a combination of three factors: the delays over the time windows, the overtime of the shift duration, and overall travel time,

There are several constraints:

(1) Each customer is visited exactly once by exactly one vehicle.

(2) All vehicle routes start at the depot and end at the depot as well.

(3) The total demand of all customers on any route cannot exceed the capacity of a vehicle. This constraint defines the feasibility of a route.

(4) The demand of any customer cannot be split.

(5) The starting time of a service cannot be earlier than the time window.

3.2 Methodology

To achieve the objective of the problem, a solution with the total costs as small as possible has to be given. The solution includes three parts – the route planning, the service scheduling, and the expected total costs (evaluation).

To provide all these parts, the methodologies are stated as follows.

(1) The Route Planning Method
The initial solution is formed by some specific route construction method. Afterwards, a Tabu Search algorithm is used for a certain number of cycles to pursue the improvements of the routes. Chapter 4 elaborates on this method.

(2) The Service Scheduling Method
When a solution is selected as the "current solution", the service schedule is to be made with an LP model. The decisions are made to minimize the expectation of the total costs possibly occurring given the solution. The derived LP model for this problem is developed in Chapter 5.

(3) The Evaluation Method
To evaluate a solution, the exact result can only be obtained from the LP model due to the complexity of the problem. However, when selecting the best solution from all candidates (i.e. the neighborhood of the current solution), it is not possible to evaluate all of them with the LP model due to the extremely long time of computation. Therefore, reasonable substitutions which

can spare the LP model are aspired to. Finally an exact evaluation function and some approximate evaluation functions are used. The details of this part are elaborated in Chapter 6.

4 Algorithm

A Tabu Search metaheuristic is used to assign each customer to a route. It has been extensively used for solving VRPs and offers the main framework of TSLPE. A related research review has been presented in Chapter 2. The algorithm of TS is described with a flow chart (Figure 4), part of which is developed from the steps stated by Finke et al. (2007).

4.1 TS Design

By nature, TS is a metaheuristic that must be tailored to the shape of the particular problem (Gendreau,M., A. Hertz, and G. Laporte, 1994). There are six basic elements of TS which are of particular significance: the initial solution, the neighborhood generation method, the neighbor evaluation method, the tabu list architecture, the aspiration criteria and the termination criteria. These elements are key factors to the success of finding good solutions. The design of TS in this study is largely inspired by Taillard (1993) and Gendreau et al. (1994).

4.1.1 Initial Solution

If the initial solution is not available beforehand, several initial solution generation methods can be found in past studies, like Solomon's I1 heuristic and modification of sweep heuristics (Cordeau, J.-F., G. Laporte, A. Mercier, 2001). In this project we use a fast and easy constructive algorithm – the Nearest Neighbor heuristic for the initial solution, such that it starts every route from the depot, by each time finding the nearest unvisited customer as long as all the restrictions are met, and then starts another tour. Finally all vehicles are assigned with a tour in this way. Figure 2 presents the procedure of the construction of one route.

4.1.2 Neighborhood Generation Method

The neighborhood of a solution contains all solutions that can be reached by moving nodes with some neighborhood generation methods. A number of neighborhood generation methods are available in literatures, including both intra-route exchange operators (e.g., 2-opt, Or-opt) and inter-route exchange operators (e.g., 2-opt*, Relocate, Exchange, CROSS-exchange, GENI-exchange). In this project two easy and cheap operators – 2-opt (see figure 3) and 2-opt* (see figure 4) operators are implemented. The size for exploring neighborhood can be either complete (i.e., all the vertices can be moved) or partial (i.e., only part of the vertices can be moved). In this project partial neighborhood is selected as candidates. More specifically, 50% of the nearest vertices are allowed to be exchanged, since the other neighbors are more unlikely to yield improvements.

4.1.3 Neighbor Evaluation Method

The algorithm works with two different objective functions. First, a simplified evaluation function (i.e. the approximation function) is used to estimate the total expected cost of the solution since computing the exact value of the objective is very costly in a stochastic setting. There are two options in selecting the current solution. In this project the one with smallest evaluation value is selected as the current solution. One may also select the one that provides the first improvement on the current value. Since infeasible solutions are also allowed as candidates for selection to help avoid local optima, the objective function includes a coefficient associated to the overload to penalize the infeasibility. The penalty coefficient is self-adjusted so that the current solutions found in a number of iterations are a mix of feasible and infeasible solutions. Second, after the best solution (i.e. the current solution) in neighborhood is selected, the value function of the LP model is calculated as part of the evaluation values of the current solutions. Chapter 6 specifies this issue.

```
Loop While (there is unvisited customer)
        Find the nearest unvisited customer;
        If total demand>capacity
               While (there is unvisited customer)
                         Find the next nearest unvisited customer;
                         If total demand<capacity
                               Include the customer into the route;
                         End if
               End while
        Else
               Include the customer into the route;
         End if
End Loop
```

Figure 2 Pseudo-code of the Nearest Neighbor heuristic

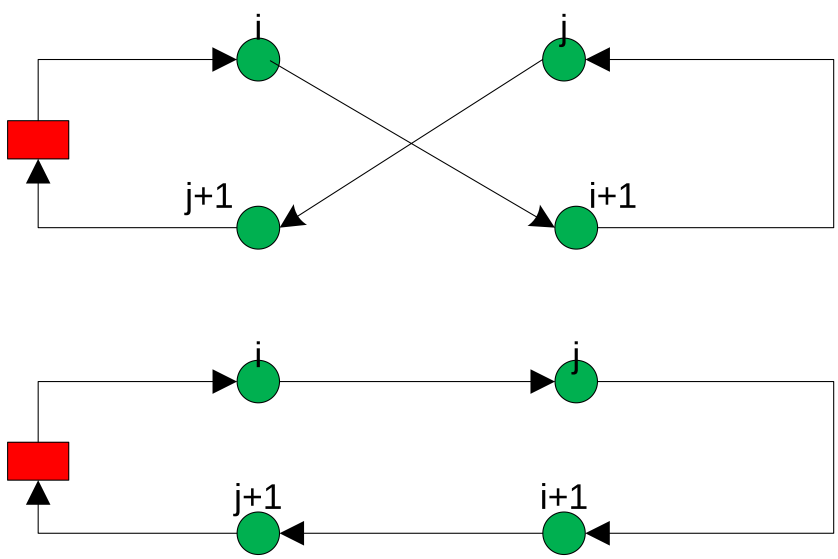


Figure 3 2-opt exchange operator

(replace edge (i,i+1) and edge (j,j+1) with edge (i,j) and edge (i+1,j+1))

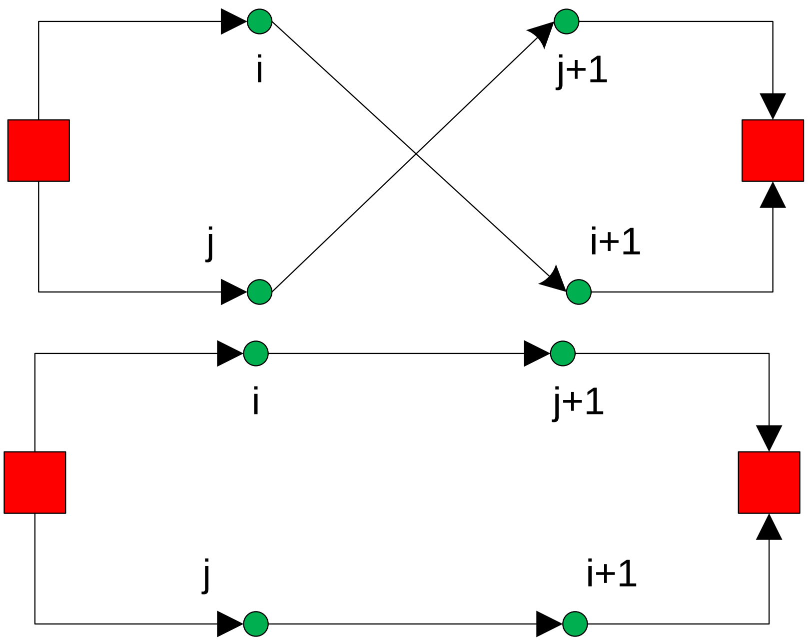


Figure 4 2-opt* exchange operator

(replace edge (i,i+1) and edge (j,j+1) with edge (i,j+1) and edge (j,i+1))

4.1.4 Tabu List Architecture

In this project at first a short-term list which is recency-based is used to avoid cycling in the search. And later, a long-term list which is frequency-based is applied as a diversification method to explore more searching space. Instead of recording full solutions, one of the attributes are usually recorded by TS designers, which can save the bookkeeping on the one hand, and at the same time avoid cycling of similar solutions on the other. The content of the tabu list can be vertices that were ever modified in some move, the pair of moves, or the move with its sequence. For the tabu list, the more specific information it holds, the less restrictive it is. In this project the pair of moves is the format stored in the tabu list. The length of the tabu list can be fixed or dynamic. In this project we set a dynamic tabu list (i.e., variable tabu tenures), which was an idea concluded to be able to reduce the probability of cycling by Taillard (1993).

4.1.5 Aspiration Criteria

An aspiration criterion is a rule that allows the tabu status to be loosed in cases where the forbidden movement exhibits desirable properties, including best so far, best in neighborhood, high influence, etc. In this project we choose to set the aspiration criterion that the tabu move which produces better result than the best solution (feasible or not) found so far is aspired from the tabu list and allowed to move.

4.1.6 Termination Criteria

When the number of non-improving moves reaches the limit or the TS has performed a certain number of iterations, then the search terminates and outputs the best configuration found so far.

4.2 Procedure

4.2.1 Steps

(1) Step 0: Initialization

Set $\gamma = \gamma_0$

$F_1^* = F_2^* = \infty$

I = 1

Decide the collection of the *N_SIZE* closest neighbors of each vertex

Clear the tabu list

(2) Step 1: Initial Solution

Generate the initial routes by searching for the nearest unvisited customer unless the route is infeasible.

Set the initial solution as the current solution *S*.

Compute $F(S)$

Update $F_1^* = F(S)$, $S^* = S$.

(3) Step 2: Neighborhood Search

Consider all candidate moves for all vertices, using certain neighborhood searching procedures.

For each move, check the following:

If the move is tabu, it is disregarded unless *S* is feasible and $F(S) < F_1^*$, or *S* is infeasible and $F(S) < F_2^*$. Otherwise, compute the value assigned using the approximated evaluation method, and the move yielding the least value is identified as the current move. In other words, set the current solution *S* with $F_a = F_a^*$.

Compute $F(S)$.

(4) Step 3: Update

Declare the current move tabu and update the tabu list

Update F_1^*, F_2^*, S^*

If F_1^*, F_2^* are improved, set *NII=0*; otherwise set *NII= NII +1*

Set I=I+1

(5) Step 4: Penalty Adjustment

When performing each searching iteration, check whether all previous *H* solutions were feasible with respect to capacity. If so, set $\gamma = \dfrac{\gamma}{B}$; if they were all infeasible, set $\gamma = B\gamma$.

(6) Step 5: Termination Check

If F_1^* and F_2^* have not decreased for the last *MAX_NII* iterations, stop.

If the total iteration number reaches *MAX_I*, stop.

The best solution is S^*, the best value is F_1^* .

Otherwise, repeat step 2.

4.2.2 Flow Chart

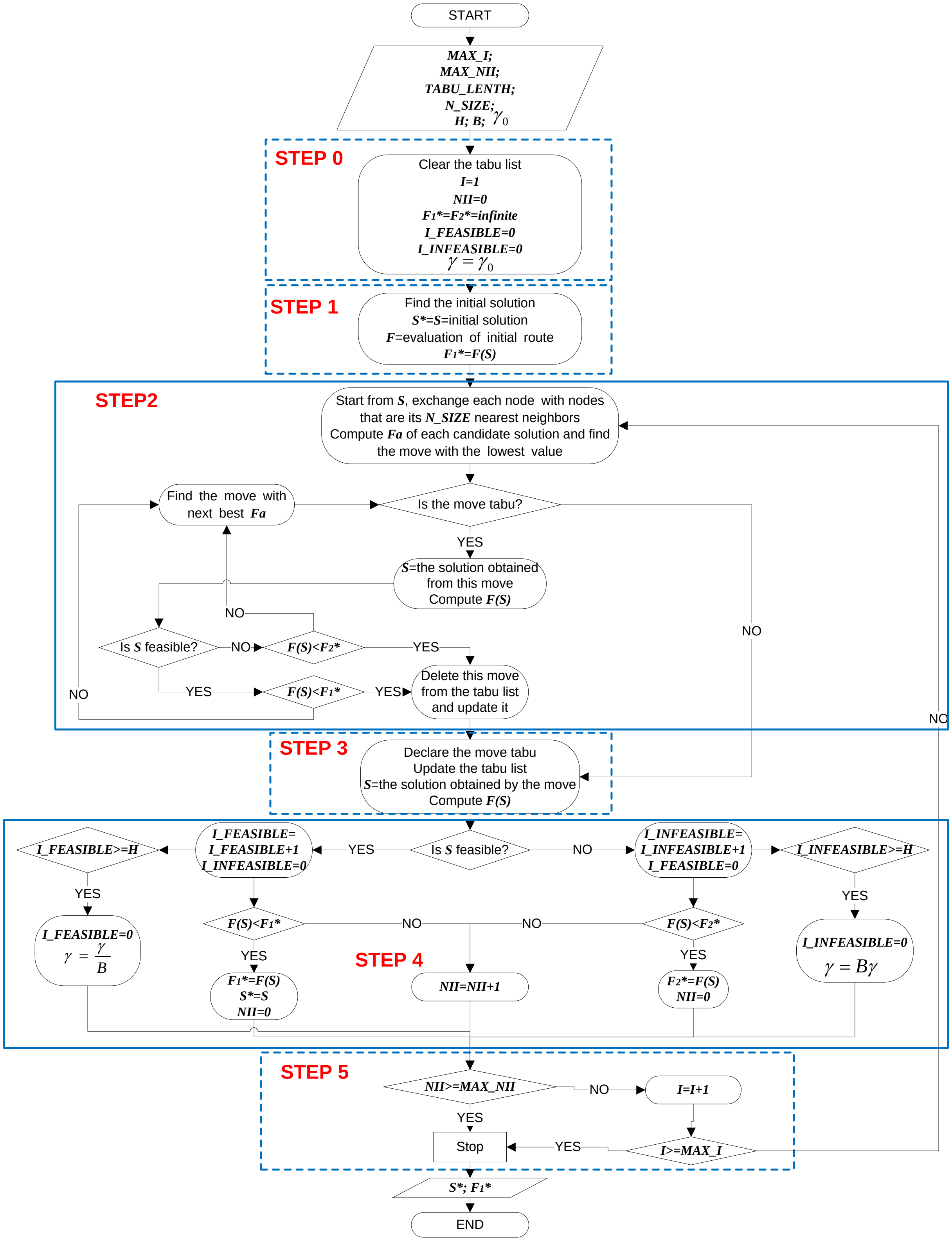


Figure 5 The flow chart of the TS algorithm

5 LP Model

5.1 Assumptions

For computational reasons, we assume that only one disruption happens at a time in each route, and all the customers and the depot have a probability to encounter a disruption. In practice the possibilities of this disruption means, for example, prolonged customer service time, or a delay in departure of vehicles. The restricted model is especially useful when disruptions are sparse and spread over time. For simplicity we assume service time for all customers to be zero and there are no disruptions during the travel (no penalty cost caused by travel). Assume unit velocity of all vehicles , so the travel time and the travel distance are equivalent.

5.2 Model Formulation

For a given route r with its all related information (customers on the route, distance, disruption distribution), we have:

Object function

$$\min \quad C_p \sum_{j\in M_r} \sum_{\substack{i\in M_r \\ i\geq j \\ i\neq 0}} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} \Delta_{rijk} + C_o \sum_{j\in M_r} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} Y_{rjk} \tag{1}$$

Subject to

$$D_{rijk} \geq 0 \quad i, j \in M_r \quad i \geq j \quad k \in \Psi_{rj} \tag{2}$$

$$D_{rijk} = d_{rjk} \quad i = j, \quad i, j \in M_r \quad k \in \Psi_{rj} \tag{3}$$

$$T_{ri} \geq t_{ri} \quad i \in M_r \tag{4}$$

$$D_{rijk} \geq D_{r,i-1,j,k} + t_{r,i-1} - T_{r,i-1} \quad i > j, \quad i, j \in M_r \quad k \in \Psi_{rj} \tag{5}$$

$$\Delta_{rijk} \geq 0 \quad i, j \in M_r \quad i \neq 0 \quad i \geq j \quad k \in \Psi_{rj} \tag{6}$$

$$\Delta_{rijk} \geq D_{rijk} - w \quad i \neq 0 \quad i \geq j \quad i, j \in M_r \quad k \in \Psi_{rj} \tag{7}$$

$$Y_{rjk} \geq 0 \quad j \in M_r \quad k \in \Psi_{rj} \tag{8}$$

$$Y_{rjk} \geq \sum_{\substack{i\in M_r \\ i\neq m_r}} T_{ri} + D_{rm_r jk} + t_{rm_r} - F \quad j \in M_r \quad k \in \Psi_{rj} \tag{9}$$

To put this model a simpler and nonlinear way for easier understanding:

Object function

$$\min \quad C_p \sum_{j\in M_r} \sum_{\substack{i\in M_r \\ i\geq j \\ i\neq 0}} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} \Delta_{rijk} + C_o \sum_{j\in M_r} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} Y_{rjk} \qquad (10)$$

Subject to

$$T_{ri} \geq t_{ri} \qquad i \in M_r \qquad (11)$$

$$D_{rijk} = d_{rjk} \qquad i = j, \qquad i, j \in M_r \quad k \in \Psi_{rj} \qquad (12)$$

$$D_{rijk} = \left(D_{r,i-1,j,k} + t_{r,i-1} - T_{r,i-1}\right)^+ \qquad i > j, \qquad i, j \in M_r \quad k \in \Psi_{rj} \qquad (13)$$

$$\Delta_{rijk} = \left(D_{rijk} - w\right)^+ \qquad i \neq 0 \qquad i \geq j \qquad i, j \in M_r \quad k \in \Psi_{rj} \qquad (14)$$

$$Y_{rjk} = \left(\sum_{\substack{i\in M_r \\ i\neq m_r}} T_{ri} + D_{rm_r jk} + t_{rm_r} - F \right)^+ \qquad j \in M_r \quad k \in \Psi_{rj} \qquad (15)$$

Explanation:

The model optimizes the schedule for a given route *r* which belongs to R_r.

The objective function (1) (or (10)) minimizes the mathematical expectation of the total penalty costs according to all the disruption scenarios and their probabilities. The first term is the weighted penalty costs to pay the customers when the arrival/starting service is beyond the bound of time window at the customer; and the second term is the weighted overtime costs to pay the drivers when the arrival at the returning depot is beyond the shift duration.

The constraints (2) and (5) (or constraint (13)) indicate that the delay time of each possibility at each customer and the depot can be either 0 (when there is no delay of arrival) or a positive value depending on the delay time and time interval previously occurred.

The constraint (3) (or constraint (12)) indicates that the delay time at a customer caused by the disruption happening on the way coming to himself is exactly equal to this disruption time.

The constraints (6) and (7) (or constraint (14)) indicate that the delay time of each possibility at each customer beyond the time window can be either 0 (when the delay does not exceed the time window) or a positive value depending on the delay time after arrival at the customer and its time window.

The constraint (4) (or constraint (11)) ensures that the time interval does not exceed the travel time.

Finally, (8) and (9) (or constraint (15)) indicate that the delay time beyond the shift duration is either 0 (when the total duration does not exceed the shift duration) or the difference between the total scheduled time duration and the shift duration (when the total duration exceeds the shift duration).

See Figure 6 for illustration.

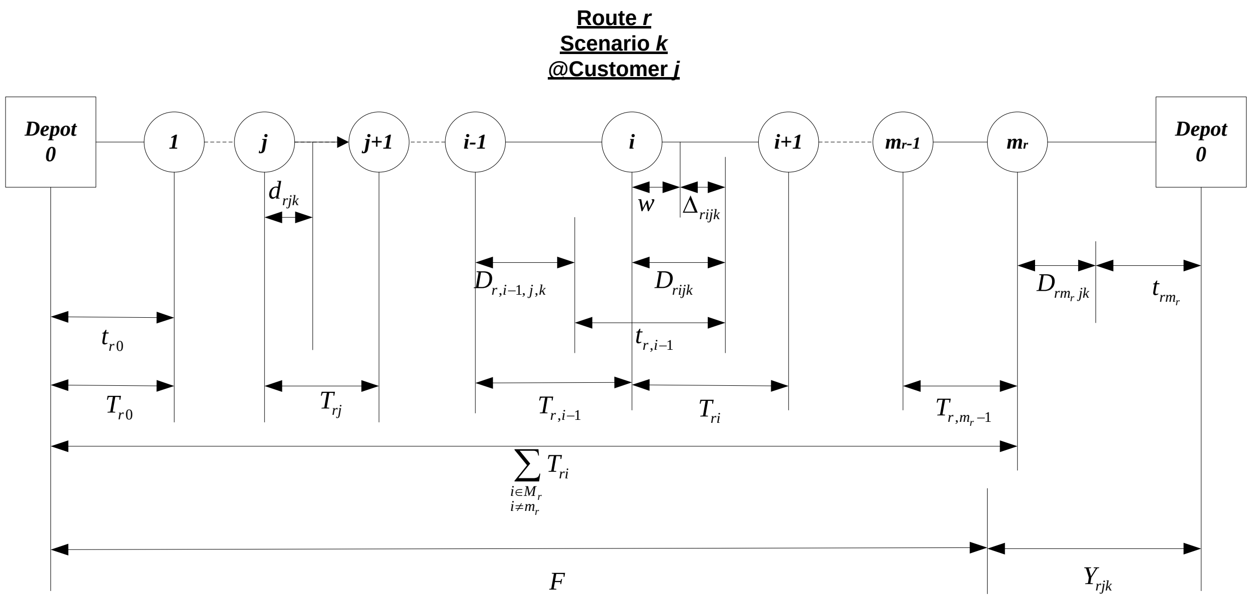


Figure 6 illustration of the LP model

6 Evaluation

With the searching procedure of Tabu Search, feasible or infeasible solutions are formed. For any solution selected, the LP model is used to determine the optimal time schedule and calculate the objective function of each route. Now that the decision making is done, the next step is the evaluation of the solutions. To evaluate a solution, the total costs of all routes should be added up.

6.1 Cost Structure

There are two kinds of costs in this problem that are of our concern.

6.1.1 Travel Cost

The travel cost is the cost taking place during the traveling process, including the cost of vehicles and energy, the labor cost, etc. It is reasonable to regard the travel cost proportional to the travel time and travel distance. Hence, in this problem we view these three concepts as equivalent. The cost rate is C_t. This cost accounts for most of the total costs and cannot be avoided. But endeavors can be made to reduce it through dexterous route arrangements.

6.1.2 Penalty Cost

Penalty costs are separated into three parts.

(1) The delay costs to pay the customers: the delay time over the time window at each customer is penalized. The cost rate is C_p.

(2) The overtime costs to pay the drivers: the overtime beyond the shift duration within which the drivers are expected to work is penalized. The cost rate is C_o. Allowing violation of the shift duration with a penalty cost happens quite often in practice.

(3) The "virtue" costs to penalize the infeasibility, i.e. the excessive demands over the capacity of the vehicles. This cost does not really take effect at the end, because the final solution must be feasible; but it is only used in the evaluation function during the intermediate searching process, to help eliminate infeasible solutions as the cost rate is adjusted. There is another option, that is, to regard the excessive demands as feasible. In this case, this cost becomes "real" and can be explained by the "lost sales", or the penalty for being unable to fulfill the customer orders. However we decide to take the excessive demands as infeasible.

6.2 Exact Evaluation

Let's call the exact evaluation function of any solution *S* (feasible or not) *F(S)*.

If we denote the objective function in the LP model of a solution *S* (infeasible or not) $F_o(S)$,

then $F_o(S) = \sum_{r \in R_r} \left\{ C_p \sum_{j \in M_r} \sum_{\substack{i \in M_r \\ i \geq j \\ i \neq 0}} \sum_{k \in \Psi_{rj}} p_{rj} g_{rjk} \Delta_{rijk} + + C_o \sum_{j \in M_r} \sum_{k \in \Psi_{rj}} p_{rj} g_{rjk} Y_{rjk} \right\}$.

For any feasible solution *S*, the total cost function is

$$F_c(S) = F_o(S) + C_t \sum_{r \in R_r} \sum_{i \in M_r} t_{ri}$$

where $C_t \sum_{r \in R_r} \sum_{i \in M_r} t_{ri}$ calculates the total travel costs of the solution

For any infeasible solution *S*, a penalty term is added, so the evaluation function is rewritten as

$$F(S) = F_c(S) + \gamma \sum_{r \in R_r} \left\{ \left[\left(\sum_{i \in M_r} q_{ri} \right) - Q \right]^+ \right\}$$

where $\gamma \sum_{r \in R_r} \left\{ \left[\left(\sum_{i \in M_r} q_{ri} \right) - Q \right]^+ \right\}$ penalizes the infeasible route which contains the total demand exceeding the vehicle's capacity

So the evaluation function is

$$F(S) = \sum_{r \in R_r} \left\{ C_t \sum_{i \in M_r} t_{ri} + C_p \sum_{j \in M_r} \sum_{\substack{i \in M_r \\ i \geq j \\ i \neq 0}} \sum_{k \in \Psi_{rj}} p_{rj} g_{rjk} \Delta_{rijk} + + C_o \sum_{j \in M_r} \sum_{k \in \Psi_{rj}} p_{rj} g_{rjk} Y_{rjk} + \gamma \left[\left(\sum_{i \in M_r} q_{ri} \right) - Q \right]^+ \right\}$$

Notice that when the solution *S* is feasible, $F(S)$ and $F_c(S)$ coincide. Therefore, we use $F(S)$ as the exact evaluation function for both feasible and infeasible solutions.

6.3 Approximate Evaluation

When the route is fixed, we want to compute the objective function by solving the LP model. However, the computation costs considerable time given the large problem size. Hence an easy-to-compute approximate evaluation function must be used to substitute it without solving the LP model.

The difficulties in solving the LP model lie in the decision making on the time schedule (i.e. the variables of time intervals, T_{ri}) which cannot be done by hand but only with the help of a computer. Hence, if we want to drop the constraints related to T_{ri} and save the LP model, to pre-decide the schedule without any intelligent calculation is worth a try. But first of all think about the role T_{ri} is playing in the evaluation and its possibilities. The essential role of T_{ri} is the buffer to deal with disruptions. Although there are time windows also playing as buffers, they may be not enough or too much, and they are fixed. Then these flexible time intervals in some sense prolong or shorten the buffers and thus make the time windows become kind of "flexible".

Two suggestions are proposed considering two extreme situations.

6.3.1 One Extreme Situation

One extreme is to make scheduling decisions such that each time interval together with the time window can cover the maximum possibility of disruption time. In other words, the time intervals are maximized and they work their best as buffers. In this case no penalty would ever happen. But the penalty term of excessive time over shift duration tend to be the largest because of too much waiting time wasted. Define $d_{\max}$ as the longest disruption time over all disruption scenarios for all customers.

$$T_{ri} = t_{ri} + d_{\max} - w \qquad r \in R_r \qquad i \in M_r \quad i \neq m_r$$

$$\begin{aligned} D_{rijk} &= \left(D_{r,i-1,j,k} + t_{r,i-1} - T_{r,i-1}\right)^+ = \left[D_{r,i-1,j,k} + t_{r,i-1} - \left(t_{r,i-1} + d_{\max} - w\right)\right]^+ \\ &= \left[w - \left(d_{\max} - D_{r,i-1,j,k}\right)\right]^+ \leq w \qquad r \in R_r, i, j \in M_r, k \in \Psi_{rj} \end{aligned}$$

$$\Delta_{rijk} = \left(D_{rijk} - w\right)^+ = 0 \qquad r \in R_r \quad i \neq 0, i \geq j \qquad i, j \in M_r \qquad k \in \Psi_{rj}$$

$$\begin{aligned} Y_{rjk} &= \left(\sum_{\substack{i \in M_r \\ i \neq m_r}} T_{ri} + D_{rm_r jk} + t_{rm_r} - F\right)^+ \leq \left(\sum_{\substack{i \in M_r \\ i \neq m_r}} \left(t_{ri} + d_{\max} - w\right) + w + t_{rm_r} - F\right)^+ \\ &= \left[\sum_{i \in M_r} t_{ri} + m_r d_{\max} - \left(m_r - 1\right) w - F\right]^+ \qquad r \in R_r \qquad j \in M_r \quad k \in \Psi_{rj} \end{aligned}$$

The evaluation function is:

$$F_{a1}(S)=\sum_{r\in R_r}\left\{C_t\sum_{i\in M_r}t_{ri}+C_o\sum_{j\in M_r}\sum_{k\in\Psi_{rj}}p_{rj}g_{rjk}\left[\sum_{i\in M_r}t_{ri}+m_r d_{\max}-(m_r-1)w-F\right]^{+}+\gamma\left[\left(\sum_{i\in M_r}q_{ri}\right)-Q\right]^{+}\right\}$$

6.3.2 The Other Extreme Situation

The other extreme is to minimize the time intervals and assign them the same value of the traveling times, which means the “buffers” would totally vanish. Once any disruption occurs, its effect is propagated till the end of the tour with the same length of time. In this case, the total penalty of delay at customers becomes the largest, while at the same time the penalty term of excessive time over shift duration tends to be the smallest because of no waiting time wasted.

$$T_{ri}=t_{ri}\qquad r\in R_r,i\in M_r,i\neq m_r$$

$$D_{rijk}=\left(D_{r,i-1,j,k}+t_{r,i-1}-T_{r,i-1}\right)^{+}=\left(D_{r,i-1,j,k}+t_{r,i-1}-t_{r,i-1}\right)^{+}=D_{r,i-1,j,k}=d_{rjk}$$
$$r\in R_r,i,j\in M_r,k\in\Psi_{rj}$$

$$\Delta_{rijk}=\left(D_{rijk}-w\right)^{+}=\left(d_{rjk}-w\right)^{+}\qquad r\in R_r\quad i\neq 0,i\geq j\qquad i,j\in M_r\qquad k\in\Psi_{rj}$$

$$Y_{rjk}=\left(\sum_{\substack{i\in M_r\\ i\neq m_r}}T_{ri}+D_{rm_r jk}+t_{rm_r}-F\right)^{+}=\left(\sum_{\substack{i\in M_r\\ i\neq m_r}}t_{ri}+d_{rjk}+t_{rm_r}-F\right)^{+}$$
$$=\left(\sum_{i\in M_r}t_{ri}+d_{rjk}-F\right)^{+}\qquad r\in R_r\qquad j\in M_r\qquad k\in\Psi_{rj}$$

The evaluation function is:

$$F_{a2}(S)=\sum_{r\in R_r}\left\{C_t\sum_{i\in M_r}t_{ri}+C_p\sum_{j\in M_r}\sum_{\substack{i\in M_r\\ i\geq j\\ i\neq 0}}\sum_{k\in\Psi_{rj}}p_{rj}g_{rjk}\left(d_{rjk}-w\right)^{+}+C_o\sum_{j\in M_r}\sum_{k\in\Psi_{rj}}p_{rj}g_{rjk}\left(\sum_{i\in M_r}t_{ri}+d_{rjk}-F\right)^{+}\right.$$

$$\left.+\gamma\left[\left(\sum_{i\in M_r}q_{ri}\right)-Q\right]^{+}\right\}$$

$$=\sum_{r\in R_r}\left\{C_t\sum_{i\in M_r}t_{ri}+C_p m_r\sum_{k\in\Psi_{r0}}p_{r0}g_{r0k}\left(d_{r0k}-w\right)^{+}+C_p\sum_{\substack{j\in M_r\\ j>0}}\sum_{k\in\Psi_{rj}}\left(m_r+1-j\right)p_{rj}g_{rjk}\left(d_{rjk}-w\right)^{+}\right.$$

$$\left.+C_o\sum_{j\in M_r}\sum_{k\in\Psi_{rj}}p_{rj}g_{rjk}\left(\sum_{i\in M_r}t_{ri}+d_{rjk}-F\right)^{+}+\gamma\left[\left(\sum_{i\in M_r}q_{ri}\right)-Q\right]^{+}\right\}$$

6.3.3 Approximate Evaluation Function

However both these two ways may not be intelligent enough to approximate the exact cost function. In real circumstances compromise should be made between these two situations. Denote the approximation of *F(S)* as $F_a(S)$. We separate different kinds of costs (delay cost and over time cost) and associate weights to the two approximations.

$F_{a1}(S)$ and $F_{a2}(S)$ have a common part – $\sum_{r\in R_r}\left[C_t\sum_{i\in M_r}t_{ri}+\gamma\left(\sum_{i\in M_r}q_{ri}-Q\right)^{+}\right]$

For $F_{a1}(S)$, there is no delay cost. Denote its overtime cost F_{a1o}. So

$$F_{a1o}=\sum_{r\in R_r}\left\{C_o\sum_{j\in M_r}\sum_{k\in\Psi_{rj}}p_{rj}g_{rjk}\left[\sum_{i\in M_r}t_{ri}+m_r d_{\max}-\left(m_r-1\right)w-F\right]^{+}\right\}$$

$$F_{a1}(S)=F_{a1o}+\sum_{r\in R_r}\left[C_t\sum_{i\in M_r}t_{ri}+\gamma\left(\sum_{i\in M_r}q_{ri}-Q\right)^{+}\right]$$

For $F_{a2}(S)$, there are both delay cost and overtime cost parts. Denote the delay cost F_{a2p} and the overtime cost F_{a2o}. So

$$F_{a2p} = \sum_{r\in R_r}\left[C_p \sum_{j\in M_r} \sum_{\substack{i\in M_r \\ i\geq j \\ i\neq 0}} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} \left(d_{rjk} - w\right)^+ \right]$$

$$F_{a2o} = \sum_{r\in R_r}\left[C_o \sum_{j\in M_r} \sum_{k\in\Psi_{rj}} p_{rj} g_{rjk} \left(\sum_{i\in M_r} t_{ri} + d_{rjk} - F \right)^+ \right]$$

$$F_{a2}\left(S\right) = F_{a2o} + F_{a2o} + \sum_{r\in R_r}\left[C_t \sum_{i\in M_r} t_{ri} + \gamma\left(\sum_{i\in M_r} q_{ri} - Q \right)^+ \right]$$

We assume the approximate evaluation has a linear relationship with all parts of $F_{a1}\left(S\right)$ and $F_{a2}\left(S\right)$. Then

$$F_a\left(S\right) = b_1 F_{a1o} + b_2 F_{a2o} + b_3 F_{a2p} + b_0 + \sum_{r\in R_r}\left[C_t \sum_{i\in M_r} t_{ri} + \gamma\left(\sum_{i\in M_r} q_{ri} - Q \right)^+ \right]$$

where b_1, b_2 and b_3 are coefficients associated with each cost part of the two approximate functions, and b_0 is a constant.

In particular, statistical analysis is executed to determine coefficients. And the validation of the linear assumption is also ensured by experiments. See Chapter 7.3.

7 Experiments

In this chapter, our approach has been tested on a few benchmark data sets. Three purposes guide the computational experiments: firstly, to determine the best strategies used in the design of TSLPE; secondly, to tune the TS parameters; and thirdly, to verify the quality of the algorithm and try to improve it. The three stages are depicted in a flow chart (see figure 7). In the following, the test problems are first introduced. Then several experiments are conducted to determine the approximation function and the neighborhood structure. Next the performance of TS procedure is improved by tuning variable parameters involved in the procedure. In addition, we identify the impact that the parameters have on the final solution using sensitivity analysis. Finally, the algorithm is improved by determining the diversification strategy and the quality of the solution is assessed. The implementation of the algorithm was coded using Visual C++ 2008 Express Edition, and exactly evaluated in the application of Gurobi Optimizer 3.0. Experiments ran on an HP Compaq 8510w Mobile Workstation with an Intel® Core™2 Duo CPU 2.50 GHz processor, and an operating system of Windows Vista™ Enterprise. Computation times are in seconds.

7.1 Test Problems

Our algorithm was tested on ten problem instances originated from Augeat et al. (1998), seven of which have a bit small sizes and the other three are larger problems. The summary of the characteristics of each instance is displayed in Table 2.

Table 2 the summary of the test problem instances

Problem Instance	Number of Customers	Number of Vehicles
n32k5	31	5
n33k5	32	5
n36k5	35	5
n37k5	36	5
n37k6	36	6
n38k5	37	5
n39k5	38	5
n53k7	52	7
n60k9	59	9
n69k9	68	9

In each problem instance, all the vertices are recorded with their identifiers, their locations defined on a Euclidean plane so that distances between arbitrary pair of vertices can be calculated, and their demands. Vertex 0 stands for the depot, and the demand of the depot is 0.

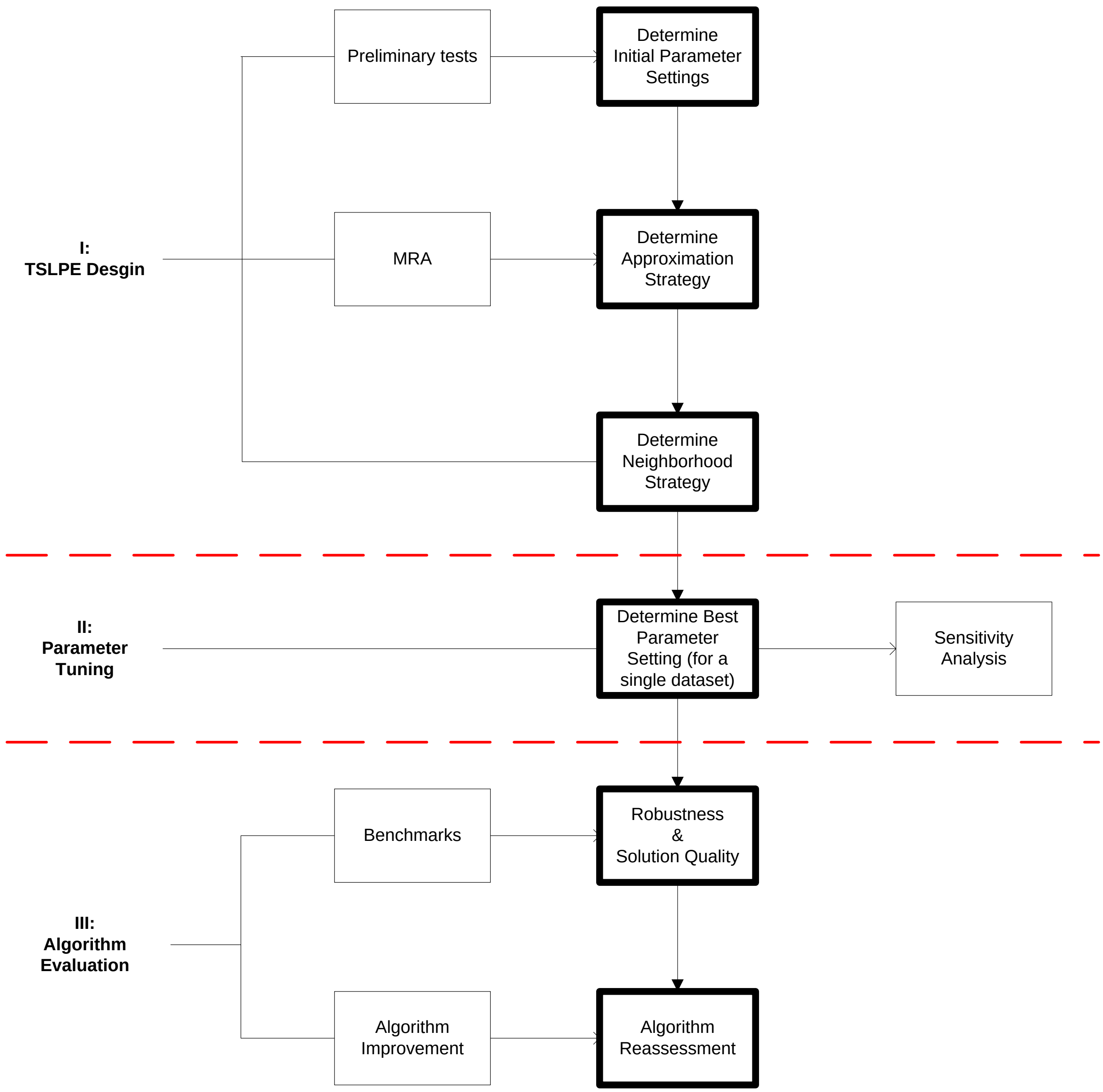


Figure 7 the procedure of the experiments

7.2 Initial Input Parameter Setting

The initial parameter setting is presented in the following tables.

7.2.1 Fixed parameters

Table 3 fixed parameter setting

Parameter	Value	Explanation
Q	100	Capacity per vehicle
F	230	Shift duration
W	30	Length of the time window
C_p	4	Penalty cost per unit of time after time windows
C_t	1	Travel cost per unit of time
C_o	5	Overtime pay per unit of time after shift duration
A	0.5	Percentage of the total nodes that determines the size of neighborhood
MAX_I	1000	Maximum number of total iteration
MAX_NII	100	Maximum number of non-improving iteration
INF	10000	A large number standing for the concept of "infinity"
N	Varies from case to case	The number of vertices (customers plus the depot)
K	Varies from case to case	The number of the vehicles (routes)

Table 4 disruption time distribution

Scenario	**1**	**2**	**3**	**4**
Probability	0.5	0.3	0.1	0.1
Disruption Time	10	20	30	60

7.2.2 Initial setting of Variables

Table 5 initial setting of variable parameters

Parameter	Value	Explanation
MAX_L	12	Upper bound of the tabu tenure
MIN_L	5	Lower bound of the tabu tenure
GAMA	4	The initial value of coefficient γ
B	3	The multiple for updating γ
H	10	The frequency for updating γ

The fixed parameter values are given for all problem sets, while the variable parameter values can be tuned at a later stage to get the solutions closer to the global optimality. *A priori* values from this project or from other researchers were selected as their initial settings. Among these parameters, initial values of MAX_L, MIN_L, GAMA, and B were obtained by some preliminary tests; they do not necessarily fit the current problems but still can provide a good start to try. And it was proposed by Gendreau et al. (1994) that it was proper to assign 10 to the parameter H.

7.2.3 Initial Approximation Function

Initial values of the coefficients mentioned in 6.3.3 are assigned with an initial guessed value 0.5, implying no preference for either of the approximation functions – F_{a1} and F_{a2}.

Hence, the initial approximation function tested to be improved is

$$F_a(S) = \frac{1}{2}F_{a1}(S) + \frac{1}{2}F_{a2}(S)$$

$$= \sum_{r\in R_r}\left\{C_t \sum_{i\in M_r} t_{ri} + \gamma\left[\left(\sum_{i\in M_r} q_{ri}\right) - Q\right]^+\right\} + \frac{1}{2}\sum_{r\in R_r}\left\{C_o \sum_{j\in M_r}\sum_{k\in\Psi_{rj}} p_{rj} g_{rjk}\left[\sum_{i\in M_r} t_{ri} + m_r d_{\max} - (m_r - 1)w - F\right]^+\right\}$$

$$+\frac{1}{2}\sum_{r\in R_r}\left\{C_p m_r \sum_{k\in\Psi_{r0}} p_{r0} g_{r0k}\left(d_{r0k} - w\right)^+ + C_p \sum_{\substack{j\in M_r \\ j>0}}\sum_{k\in\Psi_{rj}} \left(m_r + 1 - j\right) p_{rj} g_{rjk}\left(d_{rjk} - w\right)^+\right.$$

$$\left. + C_o \sum_{j\in M_r}\sum_{k\in\Psi_{rj}} p_{rj} g_{rjk}\left(\sum_{i\in M_r} t_{ri} + d_{rjk} - F\right)^+\right\}$$

7.3 Determination of the Approximation Strategy

In Section 6.3 we have discussed the necessity of substituting the evaluation function with an approximation function. Two situations from which derive the two functions F_{a1} and F_{a2} and they seem contribute two extreme decisions between which one would possibly make with the LP model. They are both assumed to be related to the exact evaluation to some extent. In this section, we use a statistical technique of Multiple Regression Analysis (MRA) to analyze the relationship between a single dependent variable and several independent variables. The objective is to use independent variables whose values are known to maximize the overall predictive power of the independent variables (Hair,J.F., W.C.Black, B.J.Babin, R.E.Anderson, 2009). The implementation of MRA is in Microsoft Office Excel 2007, and is following the steps described in the sections below.

7.3.1 The Introduction of MRA

Before we apply MRA, a few issues should be concerned:

(1) The appropriateness for the problems to apply MRA method: four rules are checked and the appropriateness to use MRA in the problems is proven. See Appendix for details.

(2) Sample size: no specific guidelines determine how large the sample size is most suitable. Rules of Thumb suggest a minimum sample of 50 and preferably 100 observations for most research situations. And the minimum ratio of observations to variables is 5:1, but the preferred ratio is 15:1 or 20:1. As a result, each experiment includes 200 observations (observations are taken every time from the 1st iteration to the 200th iteration in a single experiment) in this case to fulfill the requirements of large enough sample size.

(3) Assumptions of MRA: there are four assumptions to fulfill in order to validate the method of MRA – linearity of the phenomenon measured; homoscedasticity; independence of the error; normality of the error distribution. These assumptions were tested respectively on all data sets

after the execution of MRA, with which the applicability of the method is supported. And the results and conclusions can be referred in Appendix.

7.3.2 Estimating the Regression Model and Assessing Overall Model Fit

(1) Regression Model

Unlike $F(S)$, F_a, F_{a1}, and F_{a2} which were defined in Chapter 6, we now split any of the functions into two parts – the penalty cost due to delay at customers (later called "Delay Part") and the penalty cost due to the overtime at the returning depot (later called "Overtime Part"). Thus, F_p, F_{a1p}, F_{a2p} stand for the Delay Parts of the evaluation function, the first approximation function and the second approximation function respectively. Similarly, F_o, F_{a1o}, F_{a2o} stand for the Overtime Parts of the evaluation function, the first approximation function and the second approximation function respectively.

Now which variables should be included in the model is analyzed. Recall the two approximation functions.

$$F_{a1}(S) = F_{a1o} + \sum_{r \in R_r} \left[C_t \sum_{i \in M_r} t_{ri} + \gamma \left(\sum_{i \in M_r} q_{ri} - Q \right)^+ \right]$$

$$F_{a2}(S) = F_{a2o} + P_{a2} + \sum_{r \in R_r} \left[C_t \sum_{i \in M_r} t_{ri} + \gamma \left(\sum_{i \in M_r} q_{ri} - Q \right)^+ \right]$$

$$F_{a2p} = \sum_{r \in R_r} \left[C_p \sum_{j \in M_r} \sum_{\substack{i \in M_r \\ i \geq j \\ i \neq 0}} \sum_{k \in \Psi_{rj}} p_{rj} g_{rjk} \left(d_{rjk} - w \right)^+ \right]$$

in which F_{a1o} and F_{a2o} are variables while F_{a2p} is a fixed value given a solution, since in our tested problems, it is assumed that the disruption distribution and the disruption scenarios do not vary from customer to customer (i.e. all customers are identicle to encounter disruptions). In other words, there is no predict power on the Penalty Part due to the special situation and the limitation of the approximation functions. However, when this assumption is loosed, F_{a2p} should be various and included in the regression model.

Hence, F_a is the dependent variables and F_{a1o}, F_{a2o} are the independent predictors.

The multiple regression equations are:

$$F_a = b_0 + b_1 F_{a1o} + b_2 F_{a2o} + e$$

The multiple regression variates are:

$$F_a = b_0 + b_1 F_{a1o} + b_2 F_{a2o}$$

where

b_0 = intercept (constant)

b_1 = regression coefficient

b_2 = regression coefficient

e = prediction error (residual)

These above four values can be determined during the experiments.

Specifying the regression model to be estimated is usually done with the method of Least Squares.
(2) Identify the outliers
Through the observations we found that the first few iterations can always improve solutions dramatically, on both Delay Part and Overtime Part. So these early observations are taken as “outliers”. Although more specialized diagnostic methods can be used to detect the outliers, they are often easily identifiable. Here the outliers are selected simply by personal diagnosis. And also since the sample size is large enough, even wrongly taking a few more observations as the outliers is not a problem. The outliers of each set of experiments are identified as in Table 5 and the figure illustrations can be referred in Appendix .
(3) Assess the statistical significance of the overall model
Significance of the overall model is tested by the Coefficient of Determination (R^2), which reflects the percentage of explained variance in the dependent variable. The value lies between 0 and 1; the closer it is to 1, the more percentage of variance is explained, thus the better the future outcomes are likely to be predicted by the model. According to rules of thumb, in physical and life sciences, at least a value 0.60 is required for R^2.

Table 6 the total sample size and outliers of each data set

Data Set	Total Observations	Outliers
n32k5	200	The first 16 observations
n33k5	200	The first 6 observations
n36k5	200	The first 26 observations
n37k5	200	The first 5 observations
n37k6	200	The first 4 observations
n38k5	200	The first 7 observations
n39k5	200	The first 5 observations
n53k7	200	The first 10 observations
n60k9	200	The first 14 observations
n69k9	200	The first 11 observations

7.3.3 Results and Analysis

When comparing regression models, the most common standard used is overall predictive fit. R^2 provides us with this information.

Table 7 the results of the regression models

Set	Coefficients			Statistics	
	b_0	b_1	b_2	R^2	Significance F
n32k5	129.3353	0.0065	0.9493	0.9838	9.1712E-163
n33k5	-24.5450	0.0314	1.9703	0.9604	2.0737E-123
n36k5	125.0459	0.0118	0.9495	0.9864	3.1983E-160
n37k5	-206.2414	0.0763	1.2008	0.9758	2.7381E-154
n37k6	-102.1116	0.0563	0.9513	0.9917	8.6743E-202
n38k5	-140.7119	0.0563	1.2632	0.9829	1.2098E-168
n39k5	-195.4873	0.0701	1.0338	0.9878	1.7051E-184
n53k7	-145.5856	0.0521	1.0251	0.9474	2.7155E-120
n60k9	-241.5826	0.0627	0.9312	0.9907	1.0147E-186
n69k9	-225.0355	0.0517	1.1195	0.9861	1.8543E-173

The coefficients of the models are determined by MRA for each set. Coefficients b_1 and b_2 are positive for all sets, implying positive relationship between either of the independent variables and the dependent variable. Values of b_1 are around 0 while values of b_2 are around 1, indicating that F_{a2o} has a much stronger relevance with F_a, as inferred by the line plots before the experiments (see Appendix). Most b_0 values are negative except in the cases of n32k5 and n36k5. Notice that when solving the models, constant parts, namely the travel costs, do not affect the prediction power and thus haven't been included in the models. Also note that the minimal possible values of F_{ao2} and F_{ao2} are 0 in theory (when there is no overtime even in radical situations). Apparently, the fact that observed b_0 values, which are obtained by making F_{ao1} and F_{ao2} equal to 0 at the same time, are mostly negative implies the fact that in most practices overtime costs cannot be avoided in all situations (especially extreme situations), as a negative total penalty cost is not possible. However, it is possible to design routing that yield 0 penalty cost, while for cases with positive b_0, the penalty cost is always positive too. Statistics indicate that the model presents good predictive power and statistical significance on all data sets. A comprehensive output report of all models is in the Appendix, with more data to support the statistical analysis. The variables line fit plots and the overall model fit plots are presented in Appendix.

Notice from preliminary tests that the variable F_{a2o} in the second approximation function has very good linear relationship with F(S) while the relationship between F_{a1o} and F(S) is much weaker (see figures in Appendix). It is natural to think about how it can perform with F_{a2} itself as the predictor.

Since the real goal of this series of experiments is to replace the LP objective function with a more computational inexpensive function while at the same time enable the searching process to reach the best candidate solution (note that when making the choice, the constant part of the function can be ignored). From this point, the function of best fit is of course good, but it is too hasty to deduce that the model with less fit does not fulfill the goal. Therefore, another set of experiments are conducted to find out their real effects on the outcome of the algorithm procedure. See Table 8 for comparisons.

Table 8 outcome of different value function substitution strategies

Data Sets	Regression Model	Only F_{a2}	Percentage of Improvement
n32k5	1340.47	1268.84	-5.65%
n33k5	770.03	757.41	-1.67%
n36k5	1086.85	1110.97	2.17%
n37k5	816.30	832.21	1.91%
n37k6	1263.72	1310.02	3.53%
n38k5	909.55	893.57	-1.79%
n39k5	1041.78	1042.08	0.03%
n53k7	1305.26	1257.69	-3.78%
n60k9	1667.4	1783.89	6.53%
n69k9	1352.96	1386.66	2.43%

The last column "Percentage of Improvement" stands for the evaluation improvement of regression model over F_{a2}. From the table, we can see that the regression model slightly excels the other strategy in these problems. Preliminary tests show that F_{a2} provides good estimation on F(S) but not better than the regression model. Yet it yields in some cases better results than the regression model. The reason may be that the alternative substitution strategy sometimes can lead to different searching space and happen to increase the diversity of solutions. In addition, it is found that none of the gaps between these two strategies is very large – they both produce close final values. As a result, however, we decide to use the multiple regression model to predict the value function in the following experiments.

7.4 Determination of Neighborhood Searching Strategy

In section 4.1.2 we have discussed the neighborhood generation methods – the 2-opt operator is used for intra-route searching and the 2-opt* operator is used for inter-route searching. However, the issue that how these two operators take action has not been discussed yet. In this section experiments are run with two different implementations: one option is that the two operators switch every five iterations and each time only one operator takes effect; the other is that the two operators are implemented at the same time at every iteration, by means of the searching following a path and judging the strategy to use at every searching node.

From Table 9, the switch strategy in most cases performs better than the combine strategy, and even if it is worse, the gap is not large. One exception is the set of n32k5 (a large gap of more than 10% emerges and the reason is not clear up to now).

In the following experiments, decision is made that a switch strategy is going to be used though.

Table 9 comparison of different neighborhood searching strategy

Data Set	Switch Strategy	Combine Strategy	% Difference
n32k5	1272.32	1139.46	-10.44%
n33k5	769.03	782.04	1.69%
n36k5	1131.09	1207.16	6.73%
n37k5	833.96	866.21	3.87%
n37k6	1291.48	1330.75	3.04%
n38k5	854.56	844.95	-1.12%
n39k5	1046.84	1121.82	7.16%
n53k7	1283	1264.86	-1.41%
n60k9	1701.52	1768.84	3.96%
n69k9	1323.31	1411.56	6.67%

7.5 Parameter Tuning

In this section, we randomly select a data set to show the procedure of tabu parameter tuning. The data set to experiment on is restricted to n36k5.

There are two sets of parameters to tune – self-adjusted penalty coefficient and the tabu tenure.

7.5.1 Self-adjusted Penalty Coefficient

Three parameters are involved.

(1) GAMA and B

From experiments we found that the parameters GAMA and B are closely related – the optimal value of B is largely dependent on the optimal value of GAMA. So the performance should be observed under the interactions between these two variables, using the initial parameter setting and the approximation function determined in section 7.3.

Table 10 results on GAMA and B

		B					
		1.5	**2**	**3**	**4**	**5**	**10**
GAMA	**1**	1178.87	1073.18	1116.49	1110.89	1081.18	1193.88
	2	1157.46	1084.29	1218.62	1156.86	1202.63	1125.08
	6	1092.04	1273.39	1083.32	1134.23	1113.3	1111.34
	10	1113.18	1143.17	1075.14	1120.95	1154.49	1102.71
	20	1139.45	1158.69	1143.62	1168.37	1086.59	1114.5

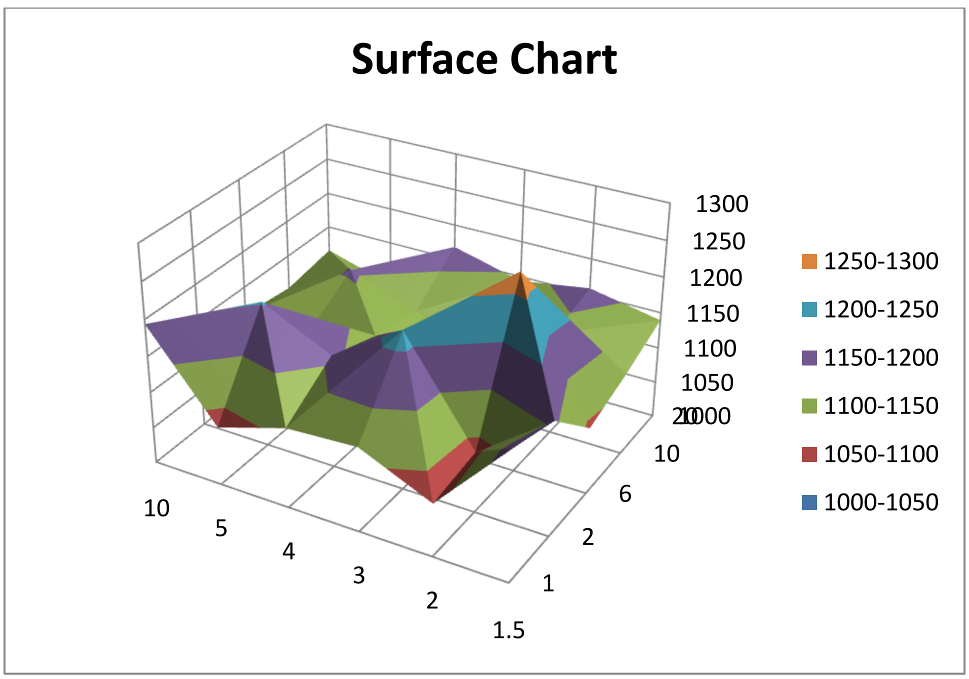


Figure 8 surface chart of GAMA and B

The choices are taken on GAMA=1 and B=2, according to the results.

(2) H

Use initial parameter setting and the approximation function determined in section 7.3, except changing GAMA and B to 1 and 2.

Table 11 results on H

H	5	8	9	10	11	12	15	18	20
value	1144.54	1099.34	1083.32	1073.18	1262.8	1062.56	1079.34	1092.88	1074.35

Hence, H=12 is selected as the optimal setting.

7.5.2 Tabu Tenure

Using initial parameter setting and the approximation function determined in section 7.3, except changing GAMA, B and H to 1, 2 and 12.

Table 12 results on tabu tenure

L_{min}, L_{max}	(5,10)	(5,12)	(11,15)	(5,15)	(8,12)	(8,15)	(16,20)	(21,25)	(21,30)
value	1111.97	1062.56	1090.49	1121.14	1088.19	1123.35	1088.19	1126.86	1122.22

Furthermore, Tabu lists with static length and dynamic length are both tested, which demonstrates the proposed preference for the dynamic tabu length.
As a result, a dynamic tabu list with length randomly assigned from 5 to 12 is decided.

7.5.3 Conclusion

The optimal (variable) parameter setting for n36k5 is

Table 13 optimal parameter setting for n36k5

Parameter	Value	Explanation
MAX_L	12	Upper bound of the tabu tenure
MIN_L	5	Lower bound of the tabu tenure
GAMA	1	The initial value of coefficient γ
B	2	The multiple for updating γ
H	12	The frequency for updating γ

7.6 Sensitivity Analysis

In this section, a series of sensitivity analysis is executed on variable parameters (i.e. GAMA, B, H, (L_{min}, L_{max})), to compare the relative importance of parameters on the final evaluation. Put another way, through the sensitivity analysis, decisions are made whether the optimal parameter setting of problem n36k5 is still robust when it turns to other different but similar problems. It also gives a hint on how much density is appropriate when adjusting parameters to find the optimal values.
The baseline values of the parameters are obtained in 7.5.3. For the ranges that values of each parameter can differ, we refer to the proposed reasonable value ranges in the literatures. other parameters remain fixed. See Table 14.

Table 14 parameter changing ranges

Parameter	Range
GAMA	1 ~ 20
B	2 ~ 8
H	5 ~ 20
Tabu Tenure	5 ~ 30

For each parameter, 4 values other than the baseline value from its suggested range are selected and results are compared. Since variables GAMA and B behave reactively, they are considered jointly. See Table 15.

Table 15 experimental data

Parameters		value	Max	Min	range	%
(GAMA,B)	(1,2)	1073.18	1086.59	1073.18	13.41	1.25%
	(2,2)	1084.29				
	(6,3)	1083.32				
	(10,3)	1075.14				
	(20,5)	1086.59				
H	5	1144.54	1144.54	1062.56	81.98	7.72%
	9	1083.32				
	12	1062.56				
	15	1079.34				
	20	1074.35				
(L_{min},L_{max})	5~12	1062.56	1126.86	1062.56	64.3	6.05%
	11~15	1090.49				
	16~20	1088.19				
	21~25	1126.86				
	26~30	1085.25				

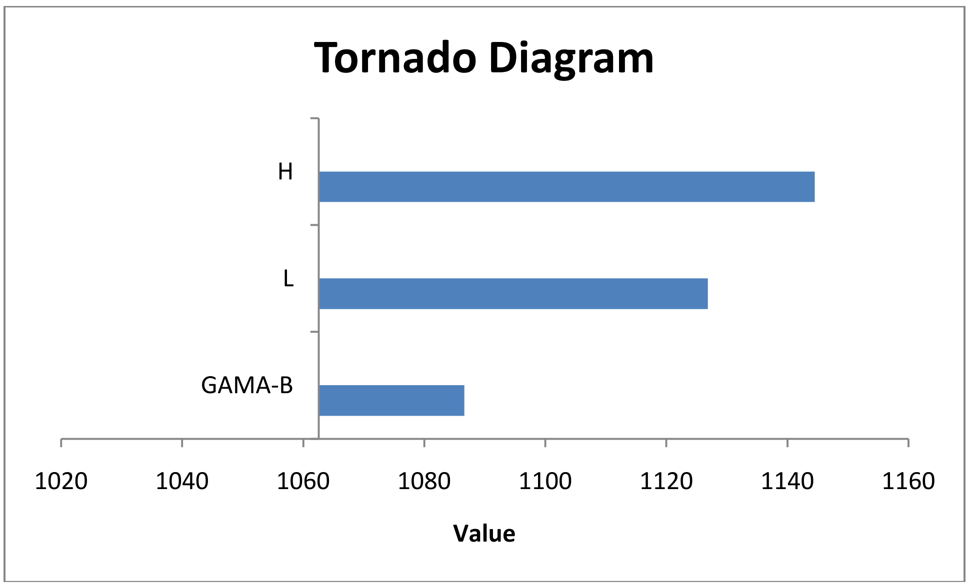


Figure 9 Tornado Diagram of sensitivity analysis

The right column computes the percentage of degree that the worst value is worse than the best value in its own category. The Tornado Diagram is a special type of Bar Chart which is useful for sensitivity analysis. From the table and the chart, all variables keep relatively low sensitivity (under 10%). This suggests these parameters may well remain the same values as optimal to sample n36k5 when testing other similar samples without costing too much more from the optimality. As the decision, after tuning the parameters for n36k5, the optimal parameter setting will be applied on other data sets too.

7.7 Solution Quality

7.7.1 Algorithm Robustness

In this following experiment, each set is tested on three evaluations – called E_1, E_2, and E_3.
To assess the algorithm, final evaluations are investigated with the penalty cost rate of its original value (C_p=4) and its double value (C_p=8).
Define $F_u(v)$ – the evaluation of the best solution found, which has the searching procedure fed by $C_p=u$ but the evaluation function fed by $C_p=v$. Both u and v have two options of values – 4 and 8.
Thus, E_1, E_2 and E_3 are defined such that when C_p=4, E_1= $F_4(4)$; when C_p=8, E_1= $F_8(8)$; when C_p=4, E_2= $F_4(8)$; when C_p=8, E_2= $F_8(4)$; when C_p=4, E_3= $F_8(8)$; when C_p=8, E_3= $F_4(4)$.
A series of experiments are conducted on the ten data sets. The results are shown in Table 16.

Table 16 self-assessment of the algorithm

Set	Initial Setting	Initial Value	E_1 [a]	E_2 [b]	E_3 [c]	Average Difference [d]	Running Time
n32k5	C_p=4	1891.38	1272.8	1407.73	1407.73	0.00%	126.5
	C_p=8	2033.4	1407.73	1272.8	1272.8		127.1
n33k5	C_p=4	1193.42	754.76	802.44	811.23	1.05%	333.4
	C_p=8	1289.26	811.23	762.48	754.76		314.9
n36k5	C_p=4	2181.04	1143.14	1261.31	1207.67	3.85%	336
	C_p=8	2309.43	1207.67	1105.85	1143.14		395.5
n37k5	C_p=4	1771.21	819.53	891.16	885.89	0.36%	511.8
	C_p=8	1908.32	885.89	820.6	819.53		562.9
n37k6	C_p=4	1949.11	1332.61	1450.77	1450.77	0.00%	202.7
	C_p=8	2101.22	1450.77	1332.61	1332.61		196.3
n38k5	C_p=4	1843.45	884.6	957.42	957.42	0.00%	433.2
	C_p=8	1963.09	957.42	884.6	884.6		458.8
n39k5	C_p=4	1884.85	1181.16	1310.23	1310.23	0.00%	246.4
	C_p=8	2059.34	1310.23	1181.16	1181.16		638
n53k7	C_p=4	1807.3	1242.89	1364.9	1364.9	0.00%	782.6
	C_p=8	1958.77	1364.9	1242.89	1242.89		767.5
n60k9	C_p=4	3050.98	1824.66	1973.08	1966.62	0.40%	540.6
	C_p=8	3285.49	1966.62	1815.9	1824.66		733.8
n69k9	C_p=4	2184.23	1351.18	1454.14	1446.78	0.44%	589.8
	C_p=8	2383.39	1446.78	1346.24	1351.18		620.2

- a: when C_p=4, E_1= $F_4(4)$; when C_p=8, E_1= $F_8(8)$.
- b: when C_p=4, E_2= $F_4(8)$; when C_p=8, E_2= $F_8(4)$.
- c: when C_p=4, E_3= $F_8(8)$; when C_p=8, E_3= $F_4(4)$.
- d: the difference (%) is calculated by $\frac{|E_2 - E_3|}{E_3} \times 100\%$ for both C_p=4 and C_p=8, and then make the average of these two percentages.

From the table, a few sets appear no difference with the two ways of starting penalty cost rates, while the others show a little bit difference. When the latter happens, it implies that the algorithm fails to reach the equally good solution when the penalty cost rate alters in a rational range; in other words, one way must outperform the other on both E_1 and E_2. Since the gap is quite small, the algorithm is thus robust to the parameter of C_p which may be a flexible parameter to determine according to various practical situations.

7.7.2 Comparison with Benchmarks

In this section, the solution quality is further tested by comparing its results with the optimal solutions solved by an exact algorithm of Branch-And-Cut (Ralphs, 2010). See Table 17.

Table 17 comparison with benchmark

Sets	Optimality (distance)	TSLPE (distance)	TSLPE/Optimality(%)
n32k5	784	1005.38	128.24%
n33k5	661	683.13	103.35%
n36k5	799	907.49	113.58%
n37k5	669	703.08	105.09%
n37k6	949	1080.36	113.84%
n38k5	730	771.60	105.70%
n39k5	822	906.77	110.31%
n53k7	1010	1040.60	103.03%
n60k9	1354	1494.57	110.38%
n69k9	1159	1201.51	103.67%
Average			109.72%

Some of the instances using our algorithm present relatively large gaps with the benchmark values (more than 10%), implying a risk that the algorithm may be not good enough. Therefore, it is necessary to seek a way to improve the algorithm.

7.8 Algorithm Improvement

A main problem of the local search is being trapped in local optima. To cope with this problem, some diversification strategies which transform one solution to a radically different solution are often added to TS. The diversification strategy is an important aspect in the design of a TS algorithm. There are many strategies proposed in related researches. They may be complicated or relatively simple, but all the main idea is to lead the searching to a different or dissimilar direction. Effective diversification is particularly supported by certain forms of long-term memory (Glover,F., and R.Mart į 2006). In this project, we will apply a diversification strategy modified from the one proposed in Taburoute algorithm by Gendreau et al. (1994). In the search, vertices that have been moved frequently are penalized by adding to the objective function of the candidate solution a term proportional to the absolute frequency of a pair of movement. In this way, a long-term memory is formed to guide the diversification procedure. This is done through the incorporation of penalties in the evaluation of movements. Taillard (1993) suggests using a constant equal to the product of three factors: a factor equal to the absolute difference value between two successive values the objective function; the square root of the neighborhood size; a scaling factor *g*. We set *g* to 0.01 as proposed by Gendreau et al. (1994). Taburoute explains the first factor into $\Delta_{\max} f_v$, where by definition in our problem $\Delta_{\max}$ is the largest observed absolute difference between the objective function values obtained at two successive iteration, and f_v is the number of times move *v* has been moved, divided by the iteration number.

Hence, the objective function *F(S)* is added by an additional penalty term. The new objective function for the diversified algorithm is:

$$F^{'}(S) = F(S) + \Delta_{\max} f_v g \sqrt{N\_SIZE}$$

Table 18 comparison between the original and diversified algorithm

Set	Algorithm	Initial Value	Final Value	%Improve[a]	Total Distance	%Improve[a]	Running Time	%Increase[b]
n32k5	original	1891.38	1272.80	14.25%	1005.38	14.99%	126.5	514.70%
	diversified		1091.47		854.71		777.6	
n33k5	original	1193.42	754.76	0.30%	683.13	0.79%	333.4	114.91%
	diversified		752.53		677.71		716.5	
n36k5	original	2181.04	1143.14	1.68%	907.49	4.11%	336.0	157.19%
	diversified		1123.88		870.19		864.2	
n37k5	original	1771.21	819.53	1.09%	703.08	1.88%	511.8	78.33%
	diversified		810.56		689.83		912.7	
n37k6	original	1949.11	1332.61	4.27%	1080.36	4.57%	202.7	315.54%
	diversified		1275.71		1030.97		842.3	
n38k5	original	1843.45	884.60	1.25%	771.60	1.37%	433.2	94.02%
	diversified		873.57		761.01		840.5	
n39k5	original	1884.85	1181.16	0.99%	906.77	-3.35%	246.4	326.95%
	diversified		1169.50		937.11		1052.0	
n53k7	original	1807.30	1242.89	0.00%	1040.60	0.00%	782.6	85.63%
	diversified		1242.89		1040.60		1452.8	
n60k9	original	3050.98	1824.66	7.07%	1494.57	6.85%	540.6	161.00%
	diversified		1695.72		1392.18		1410.9	
n69k9	original	2184.23	1351.18	0.29%	1201.51	-0.21%	589.8	192.47%
	diversified		1347.21		1204.00		1725.0	

- a: the percentage of improvement of the diversified algorithm over the original algorithm
- b: the percentage of added running time of the diversified algorithm over the original algorithm

From the table, results of all sets but one prove the advantage of the diversified strategy over the original algorithm and only one set (n53k7) remains the same solution. The computing time has different degrees of increase on different sets with the diversified strategy, but the duration is still within a reasonable and acceptable range.

Therefore, we can conclude that implementing this diversification strategy can improve the algorithm.

Table 19 comparison with benchmarks

Sets	Benchmark (distance)	TSLPE (distance) - original		TSLPE (distance) - improved		%Improvement
n32k5	784	1005.38	128.24%	854.71	109.02%	19.22%
n33k5	661	683.13	103.35%	677.71	102.53%	0.82%
n36k5	799	907.49	113.58%	870.19	108.91%	4.67%
n37k5	669	703.08	105.09%	689.83	103.11%	1.98%
n37k6	949	1080.36	113.84%	1030.97	108.64%	5.20%
n38k5	730	771.60	105.70%	761.01	104.25%	1.45%
n39k5	822	906.77	110.31%	937.11	114.00%	-3.69%
n53k7	1010	1040.60	103.03%	1040.60	103.03%	0.00%
n60k9	1354	1494.57	110.38%	1392.18	102.82%	7.56%
n69k9	1159	1201.51	103.67%	1204.00	103.88%	-0.21%
Average			109.72%		106.02%	3.70%

From the table, we can see that after improving the algorithm, it generates longer distances on all sets than the benchmark. Most sets have small gaps with the benchmark. The improved TSLPE performs well on most sets, and mostly obtains a reduced total distance from the original TSLPE. One exception is the set n39k5, whose total distance has a bit large gap with the benchmark value (14%), and is worse than the original algorithm. However, this does not necessarily indicate our new algorithm is not good. Firstly, the benchmark problem is different from our problem – it is only capacitated VRP and does not take time windows, penalty cost and overtime cost into consideration. Secondly, out problem is more complicated considering penalty cost and overtime cost whose value and searching procedure can be affected in many ways. Hence, it is understandable that longer distances are probably caused by more constraints in our problem. In our problem, shorter distance does not always yield less total costs. For example, the two optimal solutions found by the two algorithms on set n39k5 are presented below.

Table 20 an example: set: n39k5

	Original Algorithm	Improved Algorithm
Final value	1181.16	1169.50
Total distance	906.77	937.11

In this example, the first experiment by the original algorithm has larger final value (1181.16), but smaller total distance (906.77). There are many other examples found during the experiments to support this point. This suggests that in our problems, the optimal total cost is not necessarily associated with optimal distance although most time it is. The standard of total distances given by the benchmark values thus is not totally justified to evaluate our metaheuristics in this case, because the main concern in our problem is the final value of the objective cost function, not the total distance. However, it can indicate that as far as the total distance is the main concern, the improved TSLPE can provide solutions quite close to the optimality.

8 Discussion and Conclusion

In this thesis we have described a methodology to solve a special class of VRP – VRP-SITW. VRP-SITW can be viewed as a mixed problem of VRPHTW and VRPSTW, with capacity constraints and penalty for violation of the upper bound but not the lower bound of the time windows and penalty for the overtime, but it is still different from the classic VRPTW in that the time windows are quoted by the route planners instead of customers. The approach is based on the most popular metaheuristics for solving VRP – the Tabu Search. An LP evaluation method is incorporated in the framework of TS, making a new proposed procedure – TSLPE. A basic TSLPE procedure is proposed first, and then is improved by a diversification strategy using a long-term memory structure. The experiments on ten benchmark problem instances pursue better configuration of the TS procedure and tuning the values of parameters related to the TS procedure. Sensitivity analysis indicates that the effects of these parameters on solution quality are not considerable. And experimental results show good performance in terms of solution quality, computational effort, and robustness over penalty cost rate changes.

The contributions of this project include aspects that rarely appear in the past literatures, at least to the author's knowledge:

(1) First studying on such a specifically defined VRP – VRP-SITW.

(2) Proposing a new method to apply the buffer allocation model which is used extensively in machine scheduling environment in VRP, and incorporate it into the process of TS which provides satisfactory solutions

(3) Presenting a practical and statistical way that is easily computed to estimate the objective function value which costs much computational effort during the mediate searching and proving its validation

(4) One advantage of the proposed TSLPE lies in its flexibility. By raising the overtime cost rate, the shift duration becomes a hard constraint on the feasibility of the routes. By raising the penalty cost rate for tardiness, hard time window cases may also be addressed. Other features can easily be handled too, such as assigning bounded vehicles, using multiple depots, allowing hierarchical routes, and so forth.

There are many perspectives that are worthy of receiving further investigation in future study.

(1) First, as the parameters of the TS have been tested, the procedures of the other essential TS factors are also worth a try, e.g., more alternative strategies of generating an initial solution, more sophisticated neighborhood exploration to enrich the neighborhood, different memory structures, different aspiration criteria, more sophisticated diversification and intensification methods, post-optimization procedures, etc. The more successful implementations of TS often make use of better initial solutions and neighborhood structures and a balance between intensification and diversification. More complicated strategies are likely to yield better solutions, but at the same time require additional computational effort. It should be remarked that the improvement of algorithms should strike a good balance between the quality of solutions and computational efficiency.

(2) Second, techniques that can speed up the search and improve the robustness and the quality of the solutions, such as multi-search meta-heuristics, or parallel computation, have been increasingly used in recent research, e.g., (Le Bouthillier,A., T.G.Crainic, 2005), and may become a direction of future research.

(3) In addition, though lying out of the boundary of this project, the other parameters which are not directly related to the algorithm itself may find their optimized combination from a broader

managerial view in order to further reduce the total costs of the whole business. For example, experiment in 7.7.1 can reflect some effects of the parameter C_p. As another example, the effect of the shift duration *F* can also be explored. Table 21 tests larger *F* of 300 and makes comparison with the original *F* of 230. From the table, we know that if the shift duration is prolonged, then: i) the initial values become smaller; ii) the final value may become smaller or larger; iii) the traveling cost may become smaller or larger; and iv) the penalty and overtime cost must become smaller. Hence, as this example, other managerial insights can be taken into the tricky decision of the three cost rates and the shift duration in similar ways.

Table 21 results with different shift duration

Set		Initial Value	Final Value	Traveling Cost	Penalty+Overtime Cost
n32k5	F=230	1891.38	1091.47	854.71	236.77
	F=300	1174.69	862.92	802.13	60.79
n33k5	F=230	1193.42	752.53	677.71	74.82
	F=300	959.47	756.44	729.44	27.00
n36k5	F=230	2181.04	1123.88	870.19	253.69
	F=300	1517.41	912.08	851.87	60.21
n37k5	F=230	1771.21	810.56	689.83	120.73
	F=300	1257.21	742.69	708.63	34.06
n37k6	F=230	1949.11	1275.71	1030.97	244.74
	F=300	1335.18	1030.64	972.93	57.71
n38k5	F=230	1843.45	873.57	761.01	112.56
	F=300	1194.74	784.90	748.82	36.08
n39k5	F=230	1884.85	1169.50	937.11	232.39
	F=300	1213.26	889.45	838.08	51.37
n53k7	F=230	1807.30	1242.89	1040.60	202.29
	F=300	1393.44	1098.25	1040.98	57.27
n60k9	F=230	3050.98	1695.72	1392.18	303.54
	F=300	2068.35	1461.69	1392.77	68.92
n69k9	F=230	2184.23	1347.21	1204.00	143.21
	F=300	1703.85	1325.33	1265.43	59.90

REFERENCE

Augerat, P., J.M. Belenguer, E. Benavent, A. Corber, D. Naddef. (1998). Separating capacity constraints in the CVRP using tabu search. European Journal of Operational Research , 546-557.

Baker, E.K. and J.R. Schaffer. (1986). Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. American Journal of Mathematical and Management Sciences , 6, 261-300.

Balakrishnan, N. (1993). Simple heuristics for the vehicle routing problem with soft time windows. Journal of the Operational Research Society , 44, 279-287.

Baldacci,R., E. Hadjiconstantinou, and A. Mingozzi. (2004). An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research , 52, 723-738.

Baldacci,R., N. Christofides, and A. Mingozzi. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming , 351-385.

Ballestin, F., Leus, R. (2008). Meta-heuristics for stable scheduling on a single machine. Computer & Operations Research , 35, 2175-2192.

Bramel, J. and D. Simchi-Levi. (1996). Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. Operations Research , 44, 501-509.

Braysy,O. and M. Gendreau. (2005a). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Science , 39 (1), 104-118.

Braysy,O. and M. Gendreau. (2005b). Vehicle routing problem with time windows, part ii: Metaheuristics. Transportation Science , 39 (1), 119-139.

Clarke,G., and J.V. Wright. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research , 12, 568-581.

Cordeau, J.-F., G. Laporte, A. Mercier. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. J. Oper. Res. Soc. , 52, 928-936.

Derigs,U., and R. Kaiser. (2007). Applying the attribute based hill climber heuristic to the vehicle routing problem. European Journal of Operational Research , 177, 719-732.

Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics , 104, 86-92.

Ergun, O. (2001). New Neighborhood Search Algorithms Based on Exponentially Large Scale Neighborhoods. Ph.D. Thesis, Massachusetts Institute of Technology.

Finke,D.A., D.J. Medeiros, and M.T. Traband. (2007). Multiple machine jit scheduling: a tabu search approach. International Journal of Production Research , 4899-4915.

Fisher,M.L., and R. Jaikumar. (1981). A generalized assignment heuristic for the vehicle routing problem. Networks , 11, 109-124.

Gendreau,M., A. Hertz, and G. Laporte. (1994). A tabu search heuristic for the vehicle routing problem. Management Science , 40 (10), 1276-1290.

Gendreau,M., G.Laporte, R.Seguin. (1996). A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers. Operations Research , 44 (3), 469-477.

Gillett,B.E., and L.R. Miller. (1974). A heuristic algorithm for the vehicle dispatch problem. Operations Research , 22, 340-349.

Glasserman, P. (1991). Gradient Estimation via Perturbation Analysis. Norwell, MA: Kluwer Academic Publishers.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & Operations Research , 13, 533-549.

Glover,F., and R.Martí. (2006). Tabu Search. In Metaheuristic Procedures for Training Neural Networks (pp. 53-70). Springer-Verlag.

Golden,B.L., A.A. Assad, and E.A. Wasil. (2002). Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy and Newspaper Industries. In a. D. P. Toth, The Vehicle Routing Problem (pp. 245-286). Philadelphia: SIAM Monographs on Discrete Mathematics and Applications.

Golden,B.L., A.A.Assad. (1989). Vehicle routing: Methods and studies. European Journal of Operational Research , 38 (1), 126-127.

Hair,J.F., W.C.Black, B.J.Babin, R.E.Anderson. (2009). Multivariate Data Analysis (7th Edition ed.). Prentice Hall.

Halse, K. (1992). Modeling and Solving Complex Vehicle Routing Problems. Ph.D. Thesis, Technical University of Denmark, Institute of Mathematical Modelling, Lyngby, Denmark.

Herroelen, W., Leus, R. (2004). The construction of stable project baseline schedules. European Journal of Operational Research , 156, 550-565.

James,T., R.César, F.Glover. (2009). Multi-Start Tabu Search and Diversification Strategies for the Quadratic Assignment Problem. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans , 39 (3), 579-596.

Koskosidis, Y.A., W.B. Powell and M.M. Solomon. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. Transportation Science , 26, 69-85.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research , 345-358.

Laporte, G. (2007). What you should know about the vehicle routing problem. Naval Research Logistics , 811-819.

Laporte, G., and Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. Surveys in Combinatorial Optimization , 147-184.

Laporte,G., and F. Semet. (2002). Classical heuristics for the capacitated VRP. In a. D. P. Toth, The Vehicle Routing Problem (pp. 109-128). Philadelphia: SIAM Monographs on Discrete Mathematics and Applications.

Le Bouthillier,A., T.G.Crainic. (2005). A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. Computers & Operations Research , 1685-1708.

Leus, R., Herroelen, W. (2007). Scheduling for stability in single-machine production systems. Journal of Scheduling , 10, 223-235.

Leus, R., Herroelen, W. (2005). The complexity of machine scheduling for stability with a single disrupted job. Operations Research Letters , 33, 151-156.

Misevicius, A. (2005). A tabu search algorithm for the quadratic assignment problem. Computational Optimization and Applications , 30, 95-111.

Mladenovic,N., and P. Hansen. (1997). Variable neighborhood search. Computers & Operations Research , 24, 1097-1100.

Naddef,D., and G. Rinaldi. (2002). Branch-and-cut algorithms for the capacitated VRP. In a. D. P. Toth, The Vehicle Routing Problem (pp. 53-81). Philadelphia: SIAM Monographs on Discrete Mathemetics and Applications.

Novoa, C. (2005). Static and dynamic approaches for solving the vehicle routing problem with stochastic demands. Ph.D. Thesis, Lehigh University, Department of Industrial and Systems Engineering, Bethlehem, PA.

Or, I. (1976). Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking. Ph.D. thesis, Northwestern University, Evanston, Illinois.

Pisinger,D., and S. Ropke. (2007). A general heuristic for vehicle routing problems. Computers & Operations Research , 34, 2403-2435.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research , 31, 1985-2002.

Ralphs, T. (2010). Retrieved from Branch Cut and Price Resource Web: http://www.branchandcut.org/

Rochat,Y., and E.D. Taillard. (1995). Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics , 1, 147-167.

Russell, R. (1977). An effective heuristic for the M-tour traveling salesman problem with some side conditions. Operations Research , 25, 517-524.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research , 35, 254-265.

Taillard, E. (1993). Parallel iterative search methods for vehicle routing problem. Networks , 23, 661-673.

Taillard, E., P.Badeau, M.Gendreau, F.Guertin and J-Y.Potvin. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science , 31 (2), 170-186.

APPENDIX

The Appropriateness of Applying MRA

The appropriateness for the problems to apply MRA method depends on four aspects.
(1) The objective of the experiments. The applications of MRA fall into two broad classes of research problems: prediction and explanation. Our regression model should have a function of prediction that involves the extent to which the regression variate can predict the dependent variable.
(2) Specification of a statistical relationship. MRA is appropriate when the researcher is interested in a statistical, not a functional, relationship. We aim mainly at finding an appropriate substitution of the value function, and care less about the practical meaning of the regression model.
(3) Selection of the dependent and independent variables. The dependent and independent variables should be properly identified before any regression is constructed, which in this case are the exact evaluation and the two approximate evaluations of the problem.
(4) The data in this problem is metric.
To sum up, it is appropriate to use MRA as a tool to determine the approximation value function which is able to well predict the exact value function.

Assumptions of Multiple Regression Analysis

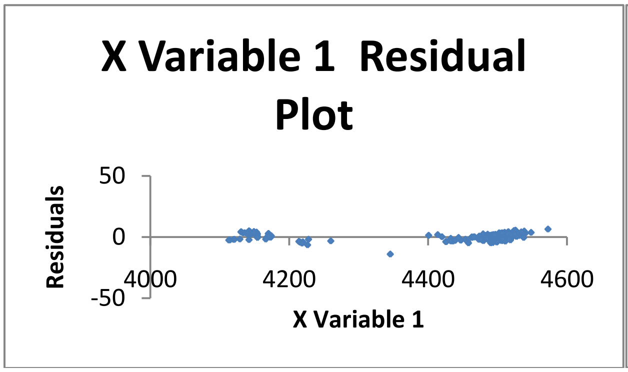
Four assumptions have to be made in order to validate the method of multiple regression analysis. They will be tested after the model is estimated.

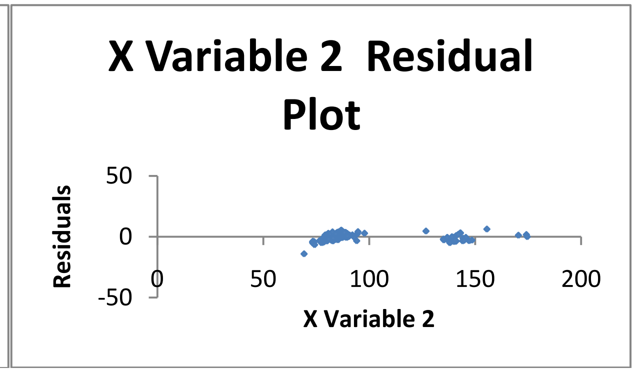
(1) Linearity of the phenomenon measured

(2) Homoscedasticity (constant variance of the error terms)

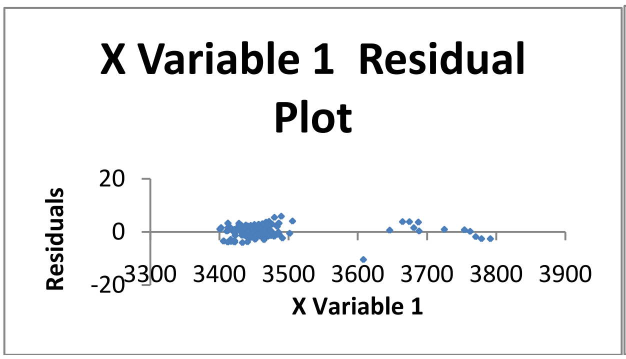
These two assumptions can be tested through residual plots of each independent variable.

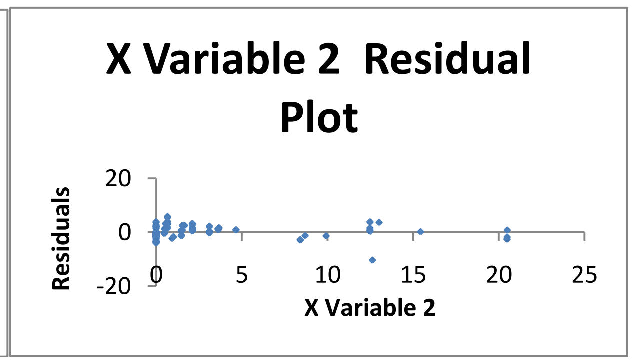
n32k5:



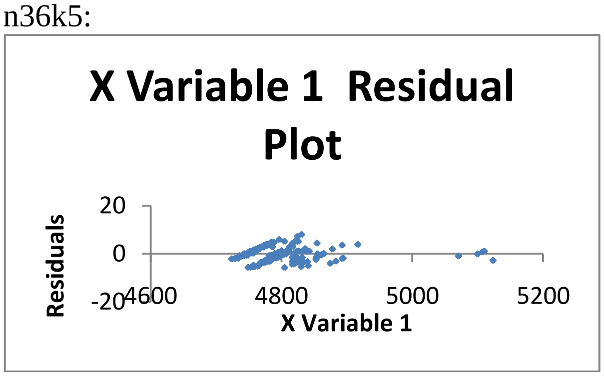


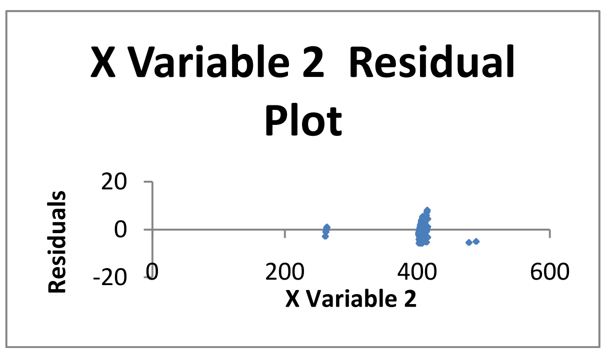
n33k5:



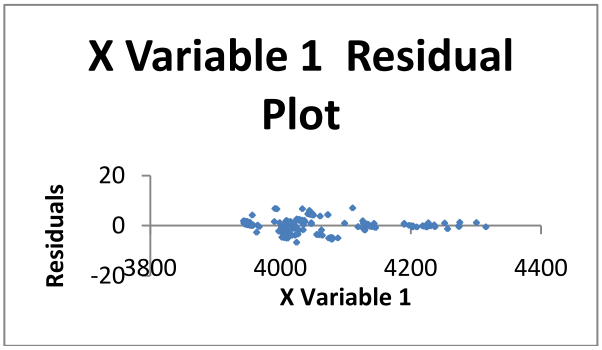


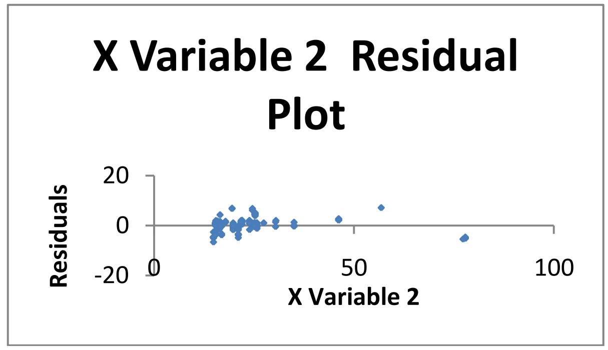
n36k5:



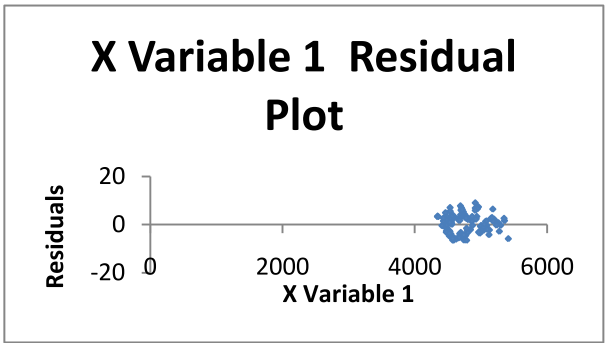


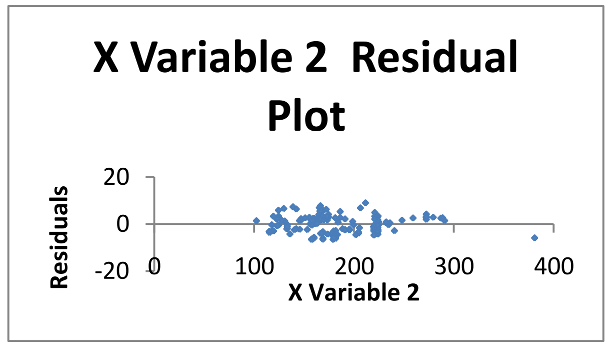
n37k5:



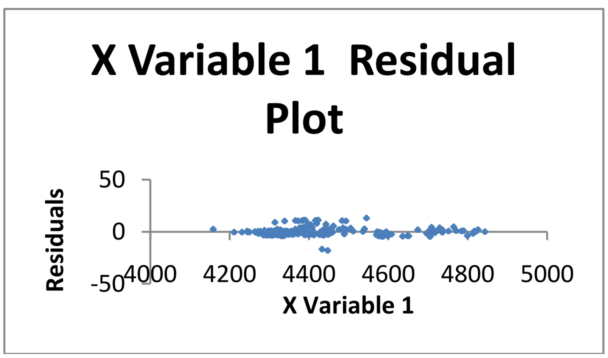


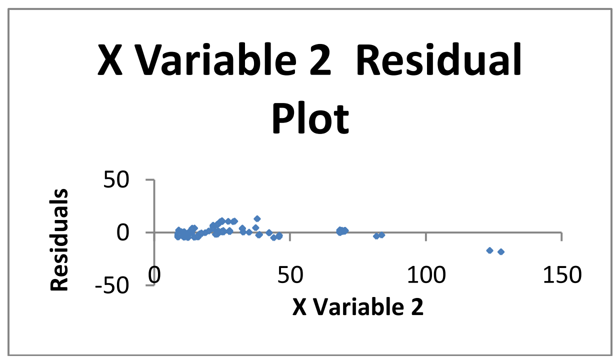
n37k6:



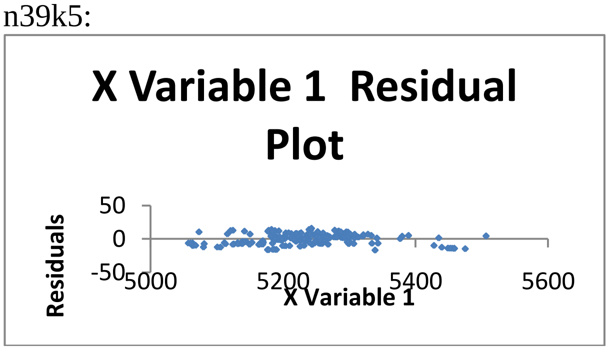


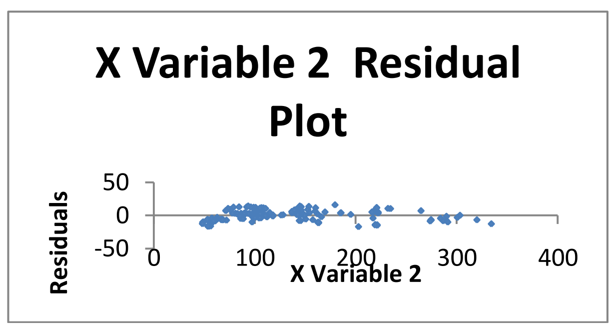
n38k5:



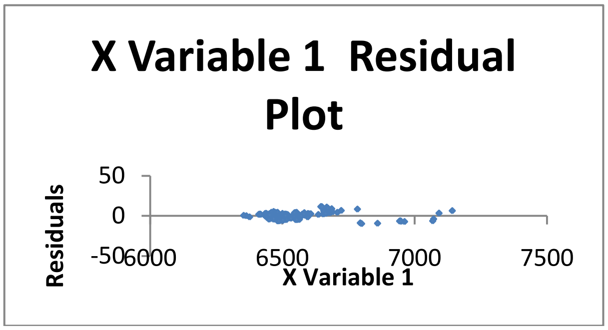


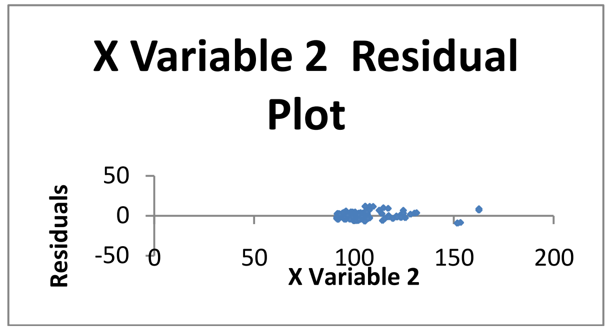
n39k5:



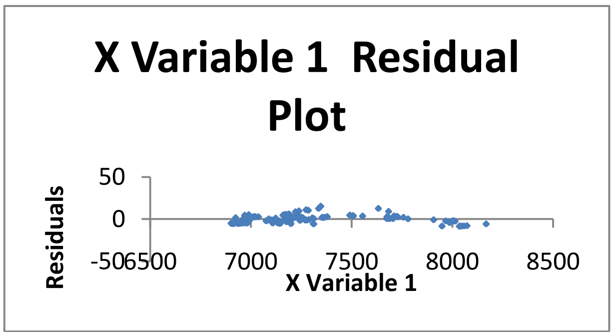


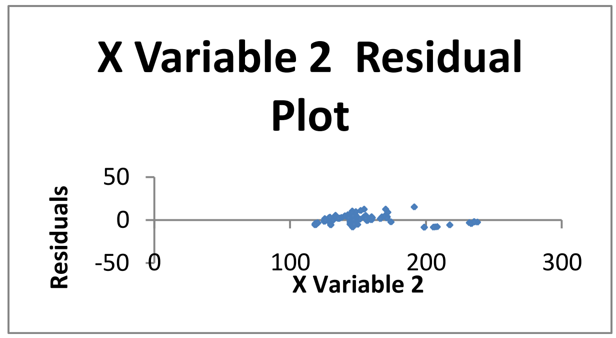
n53k7:



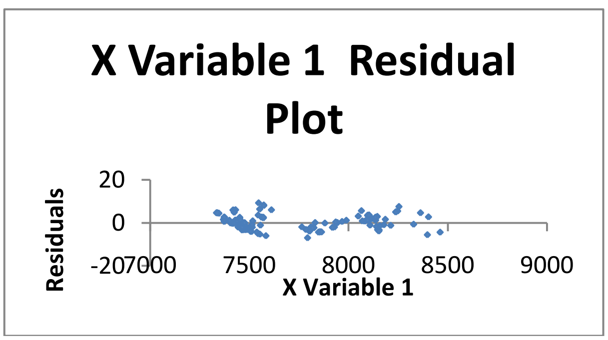


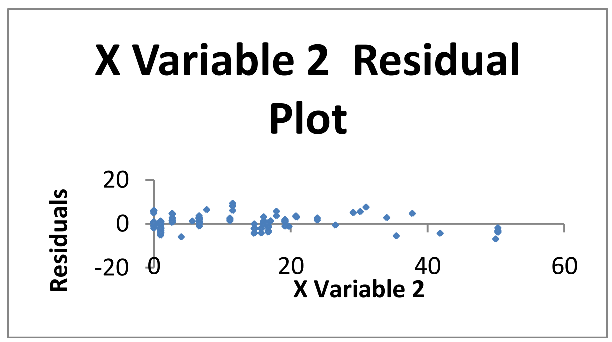
n60k9:





n69k9:

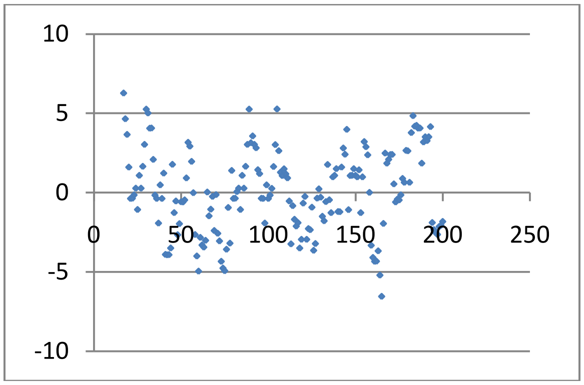




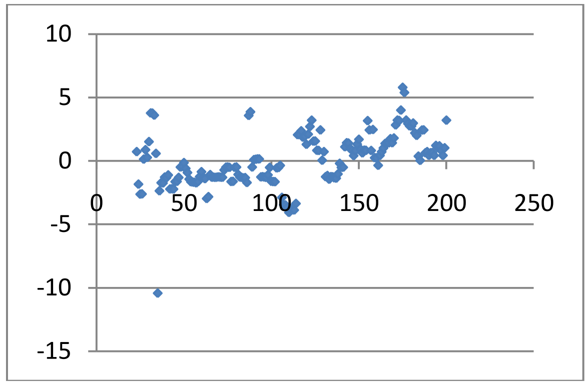
From the figures, almost all data sets meet the assumption of linearity but some may appear a little heteroscadasticity, e.g. n33k5, n36k5.

(3) Independence of the error terms. We assume in regression that each predicted value is independent, which means that the predicted value is not related to any other prediction.

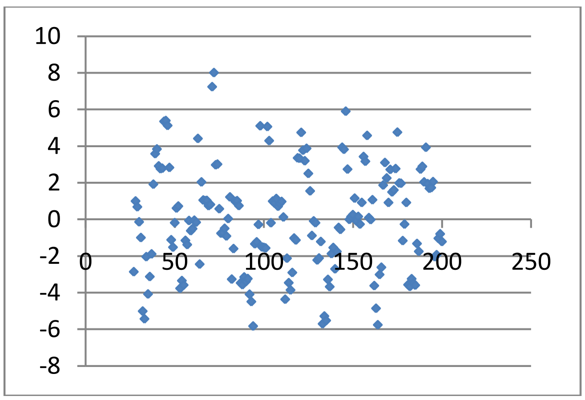
n32k5:



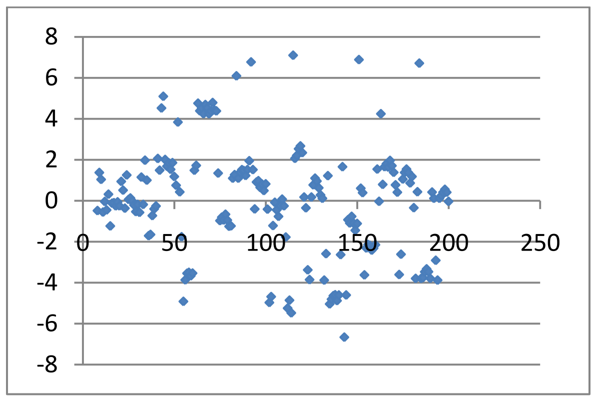
n33k5:



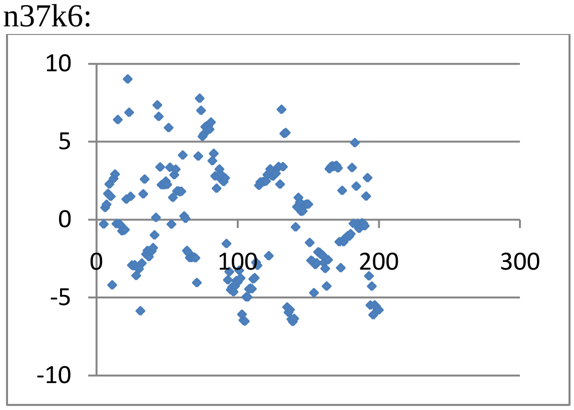
n36k5:



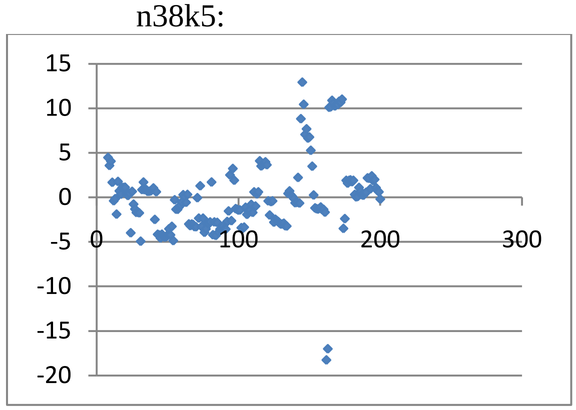
n37k5:



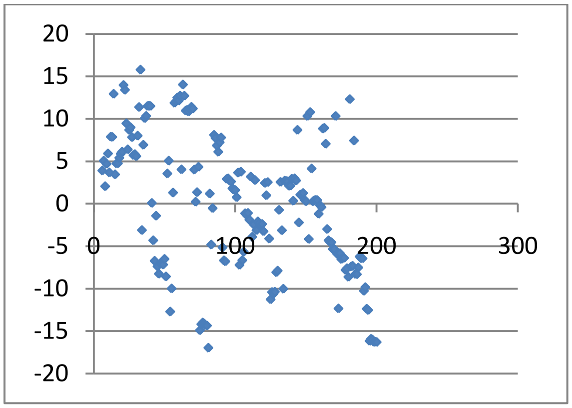
n37k6:



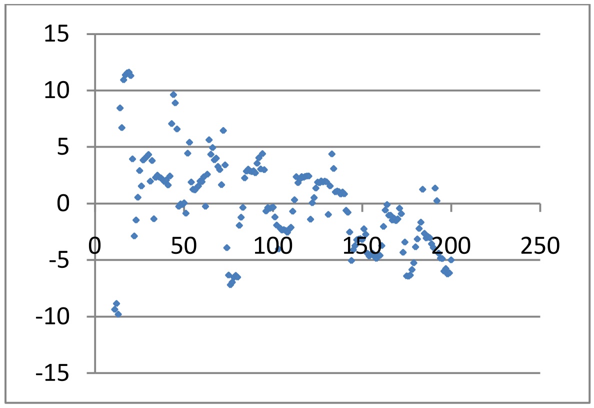
n38k5:



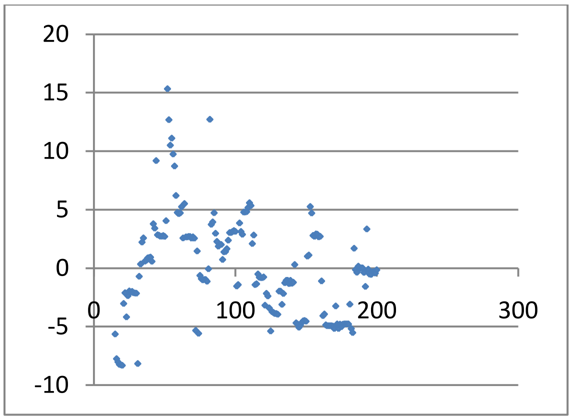
n39k5:



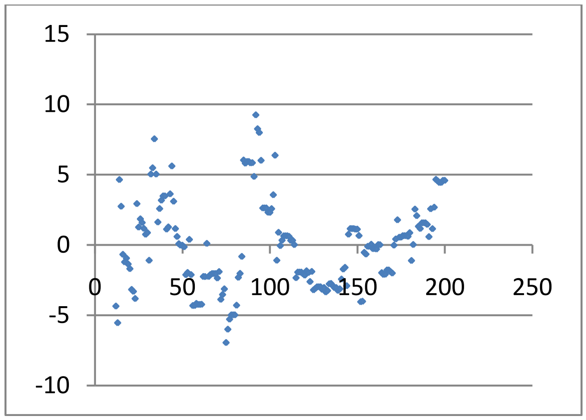
n53k7:



n60k9:



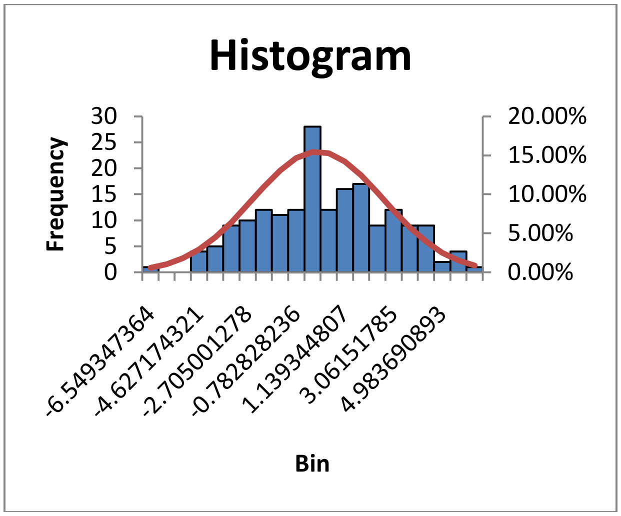
n69k9:



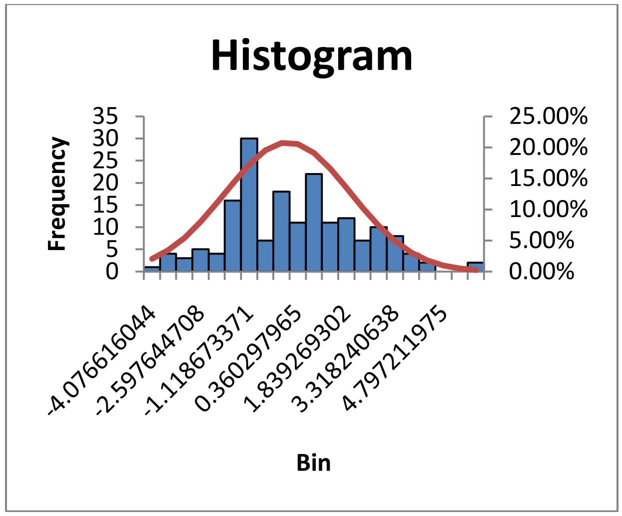
Generally speaking, the data is independent with time.

(4) Normality of the error term distribution

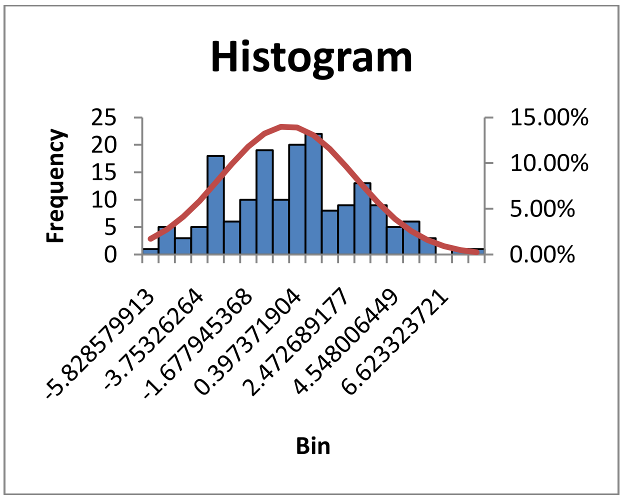
n32k5:



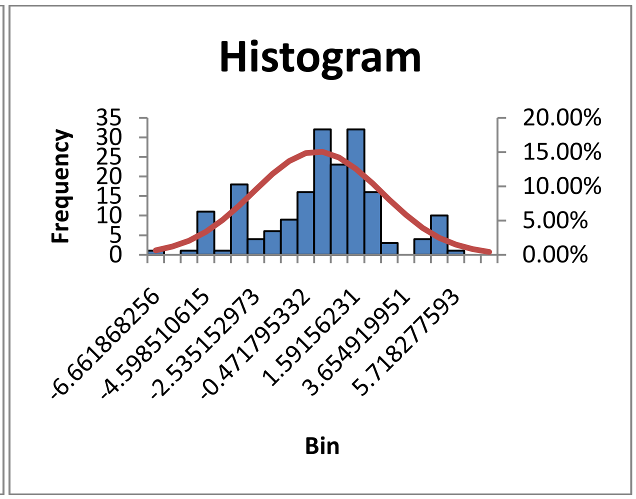
n33k5:



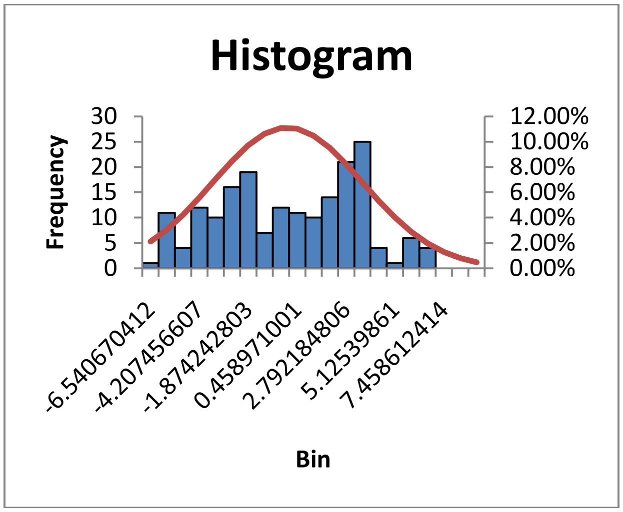
n36k5:



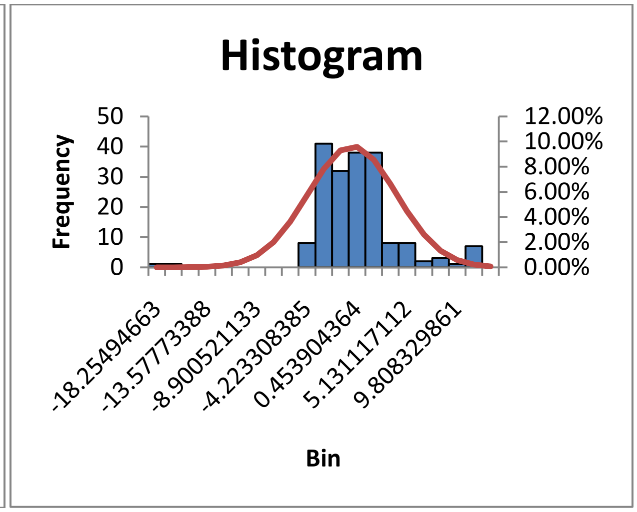
n37k5:



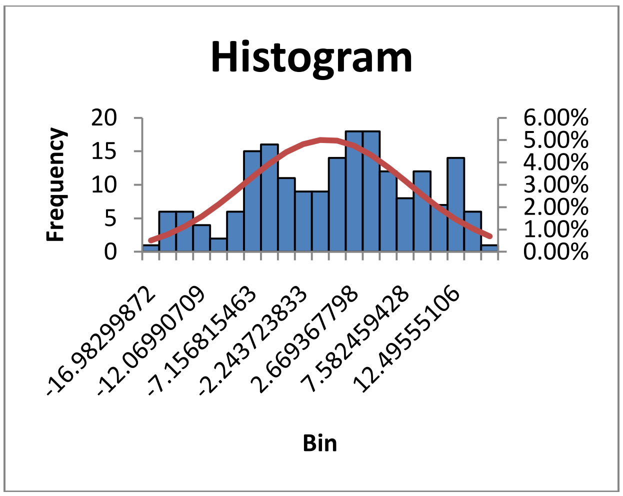
n37k6:



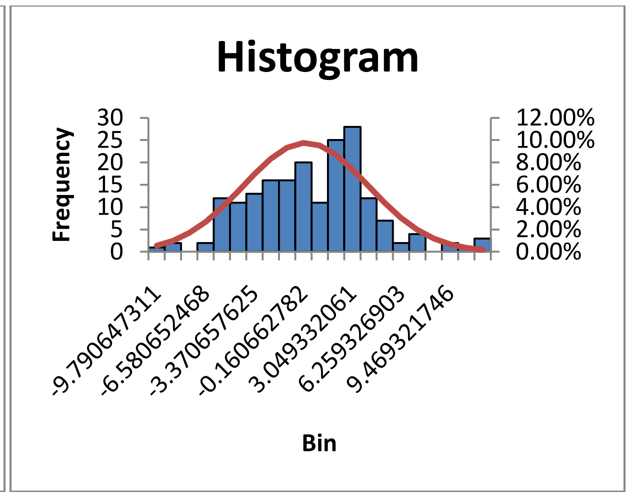
n38k5:



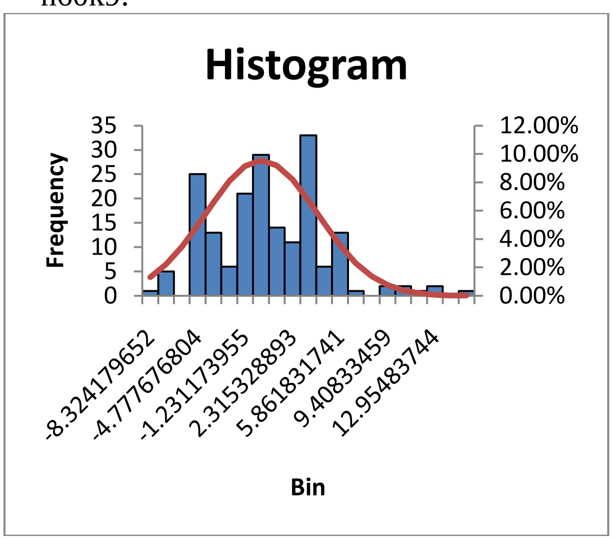
n39k5:



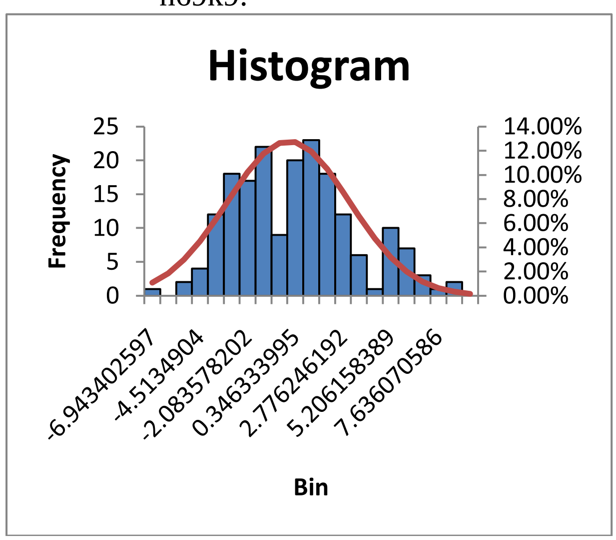
n53k7:



n60k9:



n69k9:



(5) Multicollinearity

Multicollinearity creates ‘shared’ variance between variables, thus decreasing the ability to predict the dependent measure. Hence high multicollinearity should be avoided.

Table 22 examine the multicollinearity of the variables

Sets	r Value	Cutoff value of r
n32k5	0.0980	
n33k5	0.8476	
n36k5	0.7187	
n37k5	0.3734	
n37k6	0.2942	
n38k5	0.1340	r=0.9
n39k5	0.4709	
n53k7	0.4208	
n60k9	0.8163	
n69k9	0.7920	

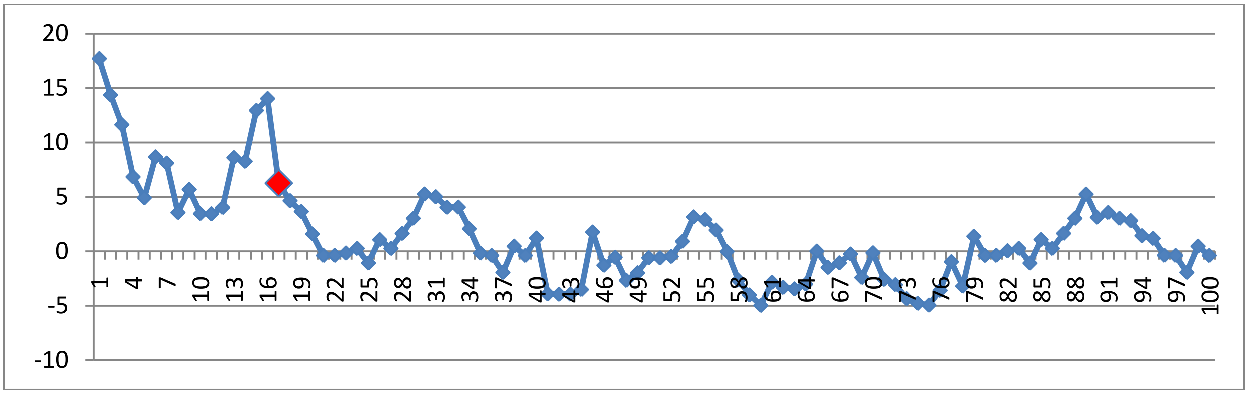
The Correlation Coefficient (*r* value) indicates the strenth of the association between any two metric variables. A common cutoff threshold is an *r* value of 0.90. The presence of higher correlations is the first indication of substantial multicollinearity. The *r* value of all data sets does not exceed the cutoff value. However, this does not ensure a lack of collinearity. It is recommended though also challenged for a researcher to determine her own acceptable degree of collinearity. At this stage, we accept the 0.90 cutoff threshold despite existence of some large *r* values (>0.80), mainly because of the irrelavence of the two variables from a theoretical (instead of statistical) point of view. Moreover, multicollinearity is harmful in reducing the overall R^2 that can be achieved, and negatively affecting the statistical significance tests of coefficient. But since the R^2 and statistical significance are satisfactory according to the experimental results, it is concluded that the effects of multicollinearity is not significant. (Hair,J.F., W.C.Black, B.J.Babin, R.E.Anderson, 2009)

As a conclusion, the MRA is suitable to be used in our problems.

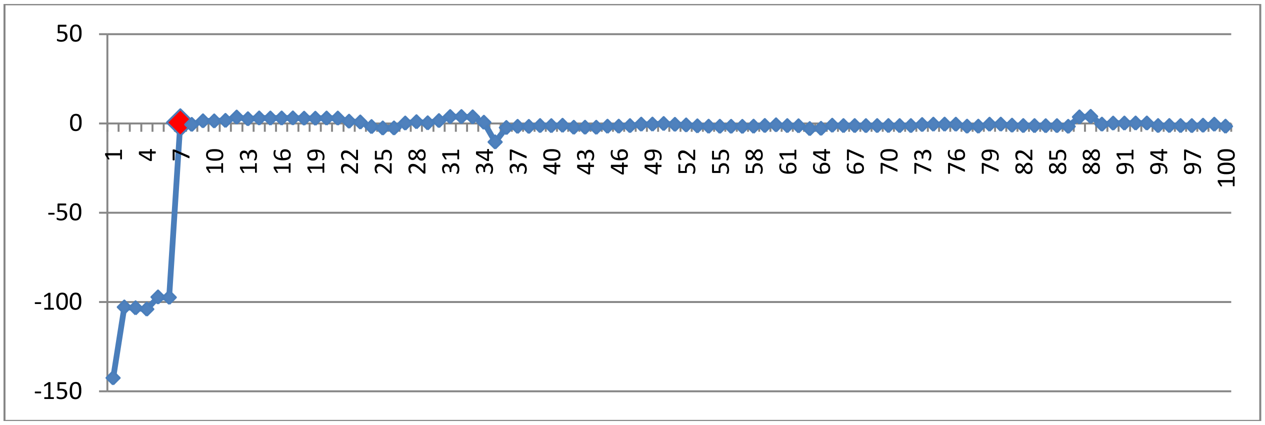
Identification of Outliers

The following figures depict the error values versus the first 100 observations. The red dots label the breaking points before which the observations are identified as outliers.

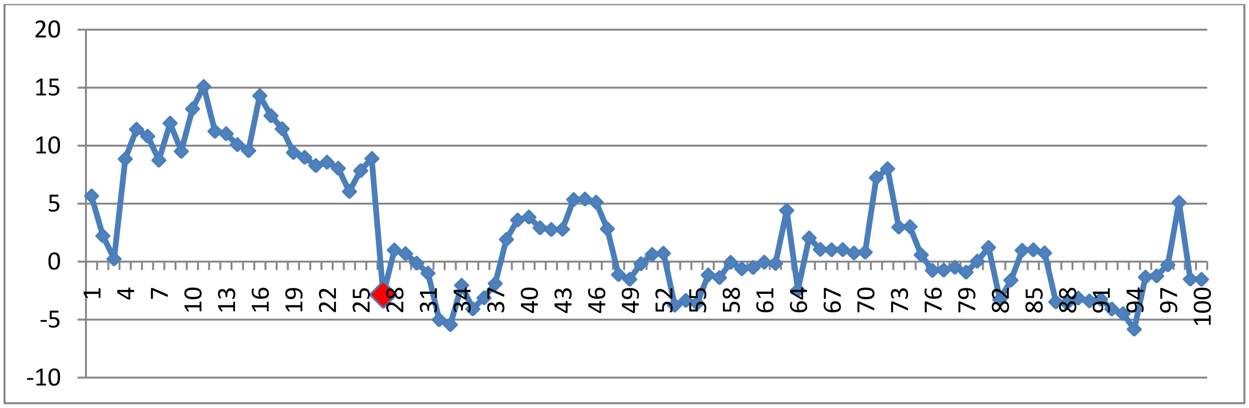
n32k5:



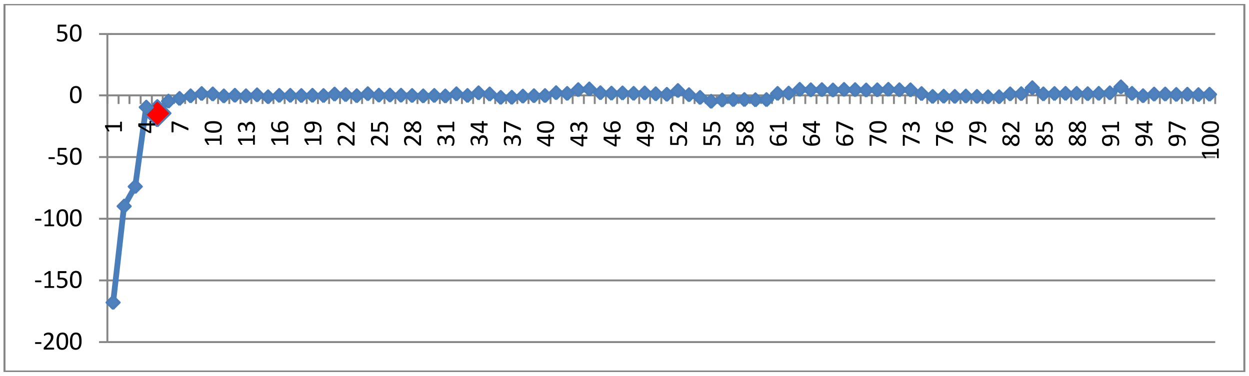
n33k5:



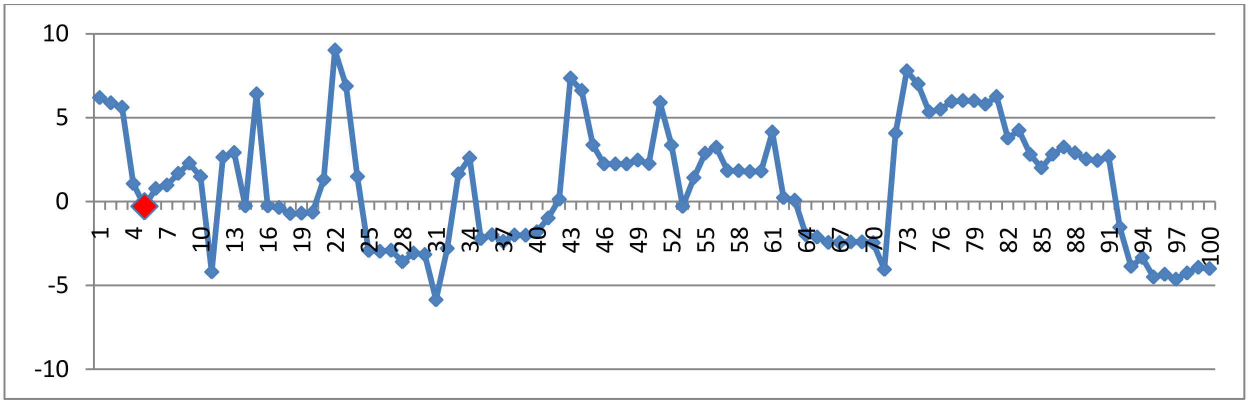
n36k5:



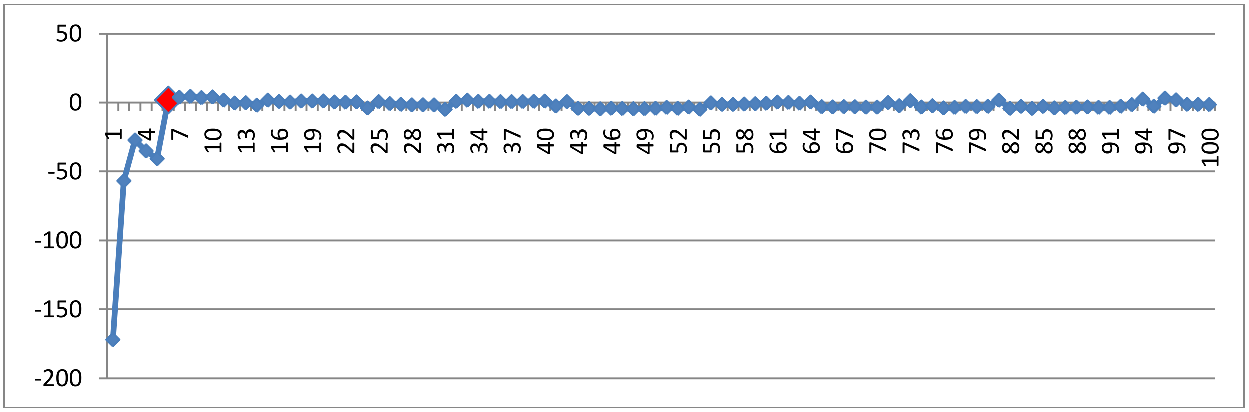
n37k5:



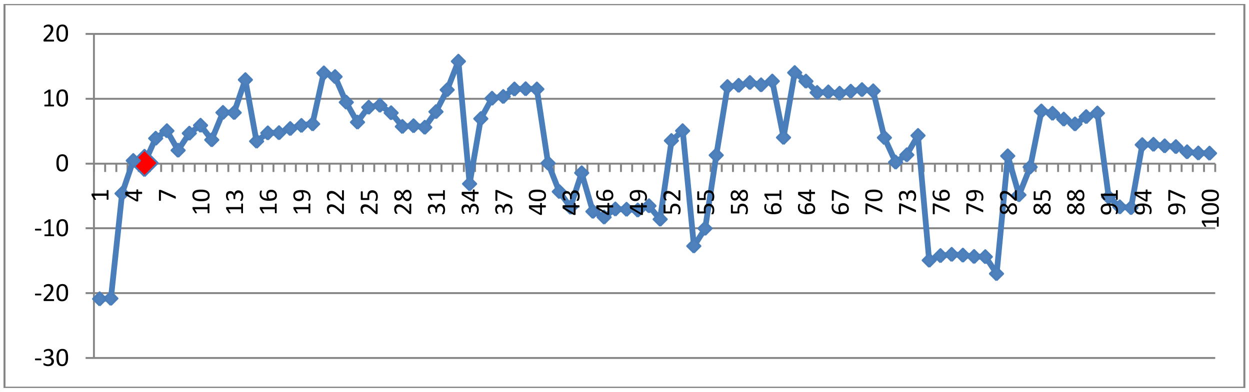
n37k6:



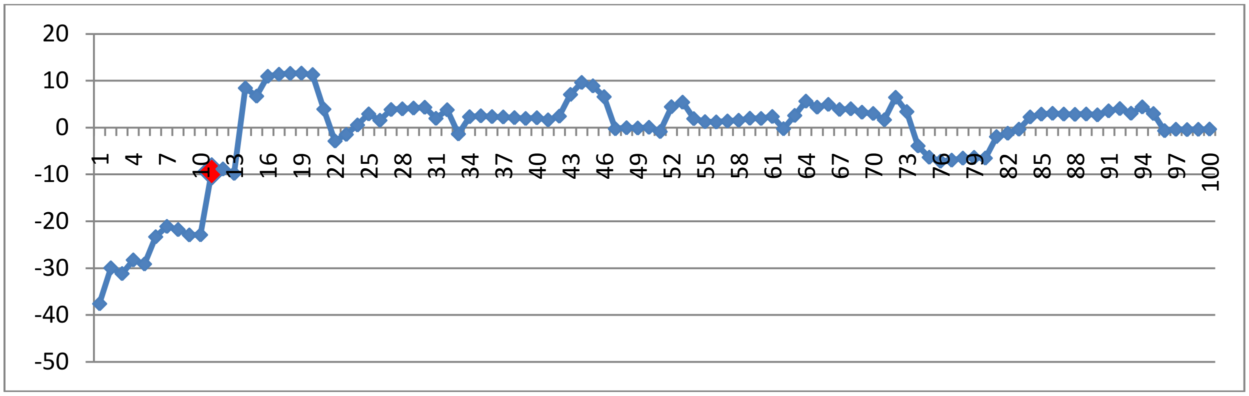
n38k5:



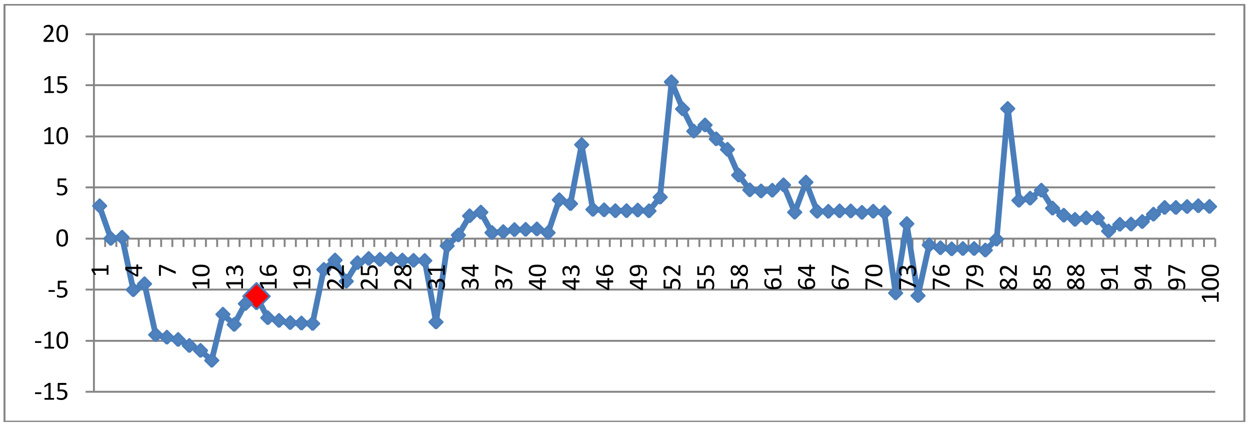
n39k5:



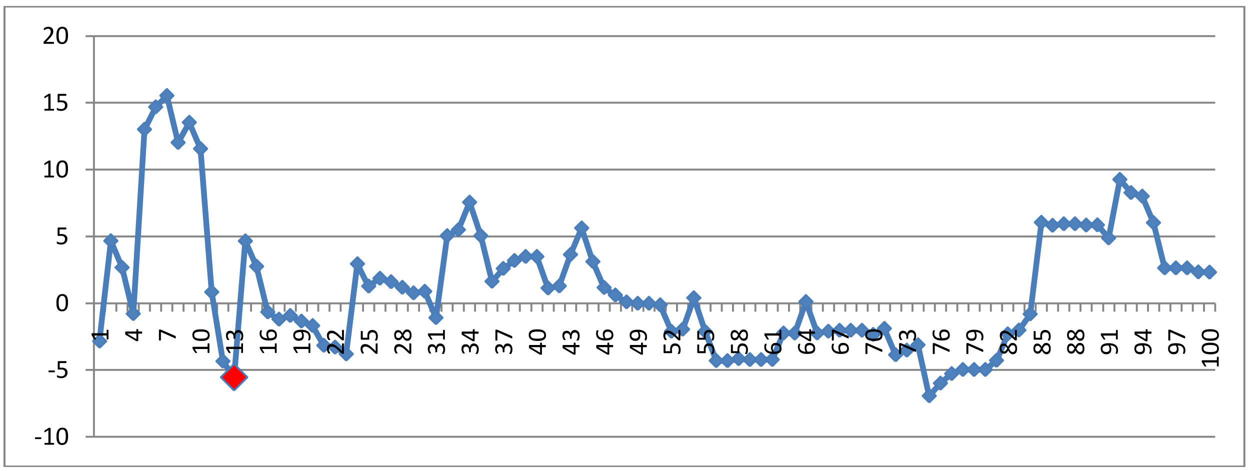
n53k7:



n60k9:



n69k9:



Summary Output of MRA

n32k5:

Regression Statistics	
Multiple R	0.9919
R Square	0.9838
Adjusted R Square	0.9836
Standard Error	2.7875
Observations	184

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	85406.2943	42703.15	5495.732	9.17E-163
Residual	181	1406.4131	7.7702		
Total	183	86812.7074			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	129.3353	6.4675	19.9976	1.05E-47
X Variable 1	0.0065	0.0015	4.4299	1.63E-05
X Variable 2	0.9493	0.0091	103.8084	3.2E-163

n33k5:

Regression Statistics	
Multiple R	0.9800
R Square	0.9604
Adjusted R Square	0.9599
Standard Error	2.0806
Observations	178

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	18363.5951	9181.7975	2121.025	2.07E-123
Residual	175	757.5652	4.3289		
Total	177	19121.1603			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-24.5450	13.8729	-1.7693	0.0786
X Variable 1	0.0314	0.0040	7.7658	6.55E-13
X Variable 2	1.9703	0.0711	27.7290	6.08E-66

n36k5:

Regression Statistics	
Multiple R	0.9932
R Square	0.9864
Adjusted R Square	0.9862
Standard Error	2.8568
Observations	174

ANOVA

	df	SS	MS	F	Significance F
Regression	2	100979.7009	50489.8504	6186.4936	3.20E-160
Residual	171	1395.5829	8.1613		
Total	173	102375.2837			

	Coefficients	Standard Error	t Stat	P-value
Intercept	125.0459	26.9794	4.6349	7.06E-06
X Variable 1	0.0118	0.0048	2.4303	0.0161
X Variable 2	0.9495	0.0120	79.0748	1.4E-136

n37k5:

Regression Statistics	
Multiple R	0.9878
R Square	0.9758
Adjusted R Square	0.9756
Standard Error	2.6634
Observations	193

ANOVA

	df	SS	MS	F	Significance F
Regression	2	54380.2954	27190.1477	3833	2.74E-154
Residual	190	1347.8029	7.0937		
Total	192	55728.0983			

	Coefficients	Standard Error	t Stat	P-value
Intercept	-206.2414	9.4304	-21.8698	8.78E-54
X Variable 1	0.0763	0.0024	32.3529	3.27E-79
X Variable 2	1.2008	0.0182	65.9836	6.3E-133

n37k6:

Regression Statistics	
Multiple R	0.9959
R Square	0.9917
Adjusted R Square	0.9917
Standard Error	3.6027
Observations	196

ANOVA

	df	SS	MS	F	Significance F
Regression	2	301126.5656	150563.2828	11600.35	8.67E-202
Residual	193	2504.9859	12.9792		
Total	195	303631.5515			

	Coefficients	Standard Error	t Stat	P-value
Intercept	-102.1116	5.6354	-18.1197	1.63E-43
X Variable 1	0.0563	0.0011	51.1926	2.99E-114
X Variable 2	0.9513	0.0063	152.1692	6.02E-203

n38k5:

Regression Statistics	
Multiple R	0.9914
R Square	0.9829
Adjusted R Square	0.9827
Standard Error	4.1645
Observations	193

ANOVA

	df	SS	MS	F	Significance F
Regression	2	189648.6527	94824.3263	5467.549	1.21E-168
Residual	190	3295.1917	17.3431		
Total	192	192943.8444			

	Coefficients	Standard Error	t Stat	P-value
Intercept	-140.7119	8.4402	-16.6715	4.87E-39
X Variable 1	0.0563	0.0019	29.6479	3.41E-73
X Variable 2	1.2632	0.0132	95.4044	1.9E-162

n39k5:

Regression Statistics	
Multiple R	0.9939
R Square	0.9878
Adjusted R Square	0.9877
Standard Error	7.9845
Observations	195

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	992497.0669	496248.5335	7783.957	1.71E-184
Residual	192	12240.5253	63.7527		
Total	194	1004737.592			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-195.4873	40.2745	-4.8539	2.50E-06
X Variable 1	0.0701	0.0078	8.9769	2.56E-16
X Variable 2	1.0338	0.0098	105.5571	5.48E-172

n53k7:

Regression Statistics	
Multiple R	0.9733
R Square	0.9474
Adjusted R Square	0.9468
Standard Error	4.1073
Observations	190

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	56786.9248	28393.4624	1683.117	2.72E-120
Residual	187	3154.6093	16.8696		
Total	189	59941.5340			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-145.5856	15.1082	-9.6362	4.26E-18
X Variable 1	0.0521	0.0024	21.2756	7.96E-52
X Variable 2	1.0251	0.0256	40.0130	1.29E-93

n60k9:

Regression Statistics	
Multiple R	0.9954
R Square	0.9907
Adjusted R Square	0.9906
Standard Error	4.2025
Observations	186

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	345247.9041	172624	9774.4506	1.01E-186
Residual	183	3231.9139	17.6607		
Total	185	348479.8181			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-241.5826	9.7864	-24.6856	3.88E-60
X Variable 1	0.0627	0.0017	37.9291	1.26E-88
X Variable 2	0.9312	0.0199	46.7675	1.02E-103

n69k9:

Regression Statistics	
Multiple R	0.9930
R Square	0.9861
Adjusted R Square	0.9860
Standard Error	3.1372
Observations	189

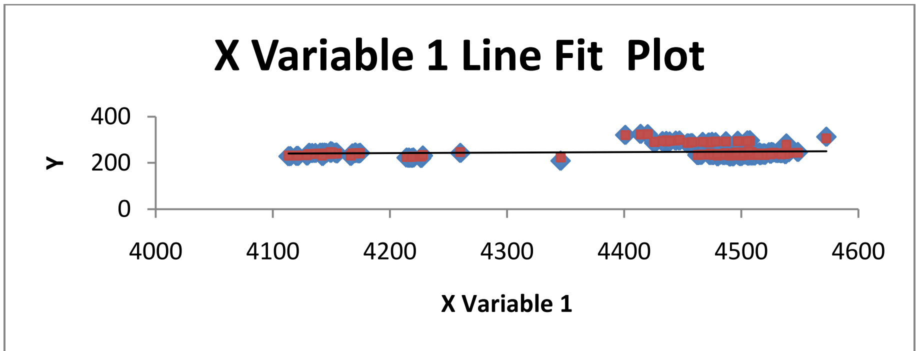
ANOVA

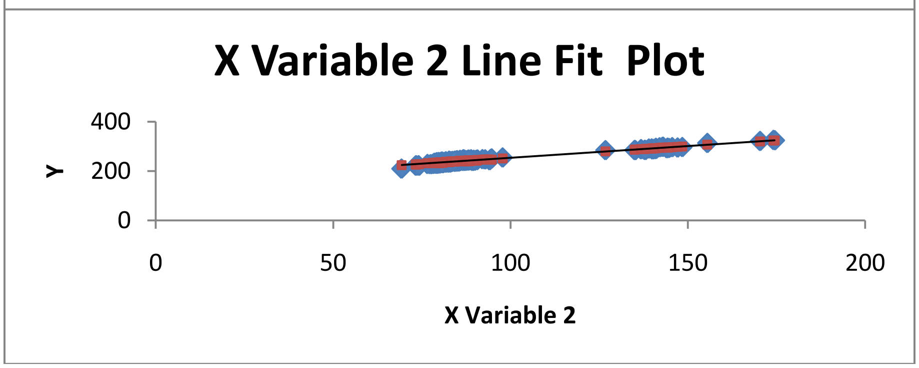
	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	100979.7009	50489.8504	6186.4936	3.20E-160
Residual	171	1395.5829	8.1613		
Total	173	102375.2837			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-225.0355	9.5217	-23.6340	6.40E-58
X Variable 1	0.0517	0.0013	40.5412	2.85E-94
X Variable 2	1.1195	0.0334	33.5410	8.45E-81

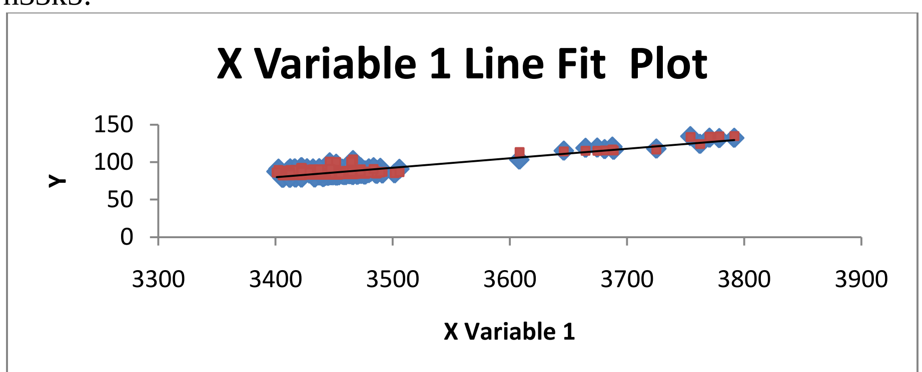
Variables Line Fit Plots

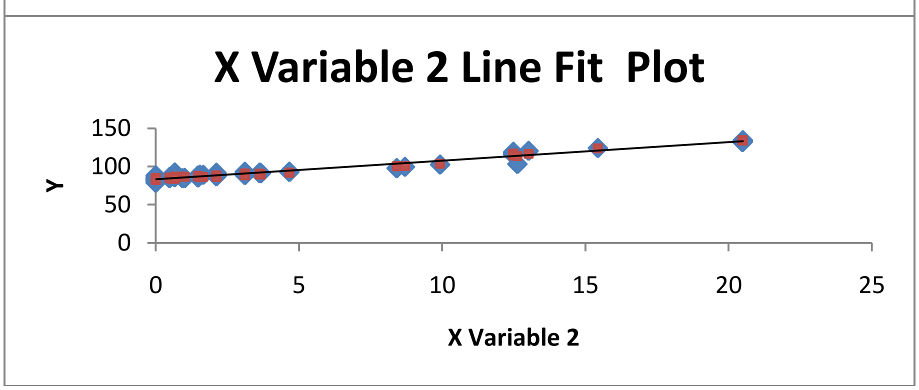
n32k5:



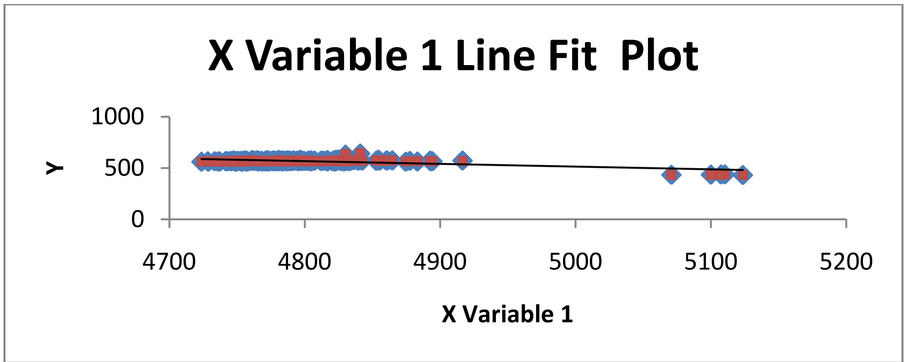


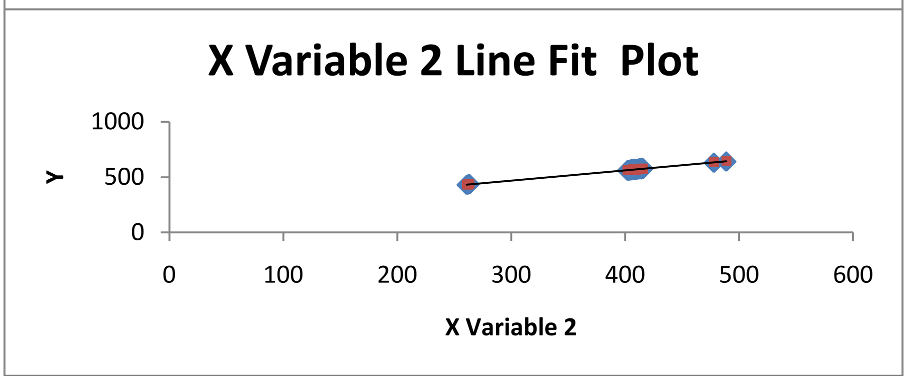
n33k5:



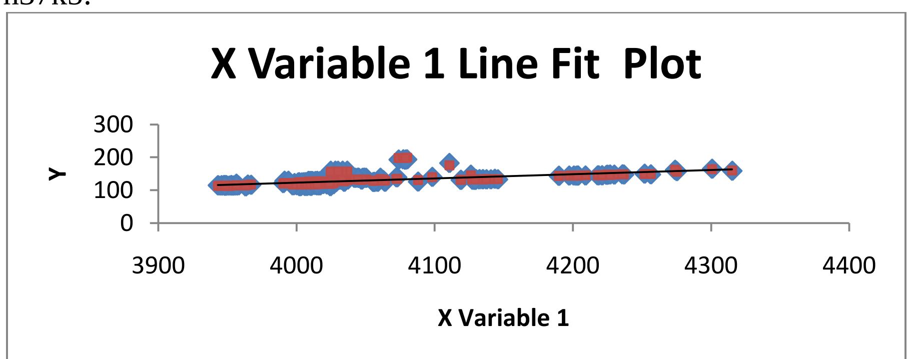


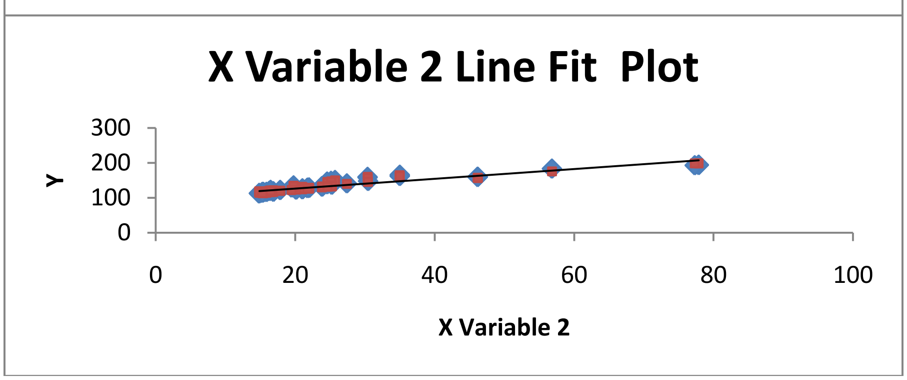
n36k5:


X Variable 1 Line Fit Plot
1000
500
0
Y
4700
4800
4900
5000
5100
5200
X Variable 1

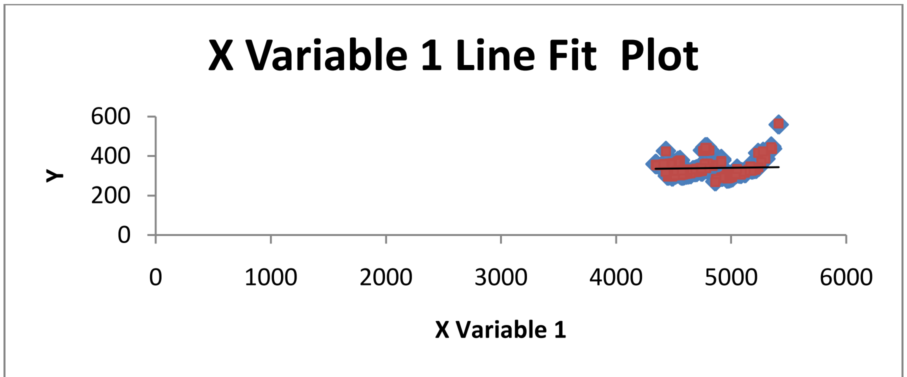

X Variable 2 Line Fit Plot
1000
500
0
Y
0
100
200
300
400
500
600
X Variable 2

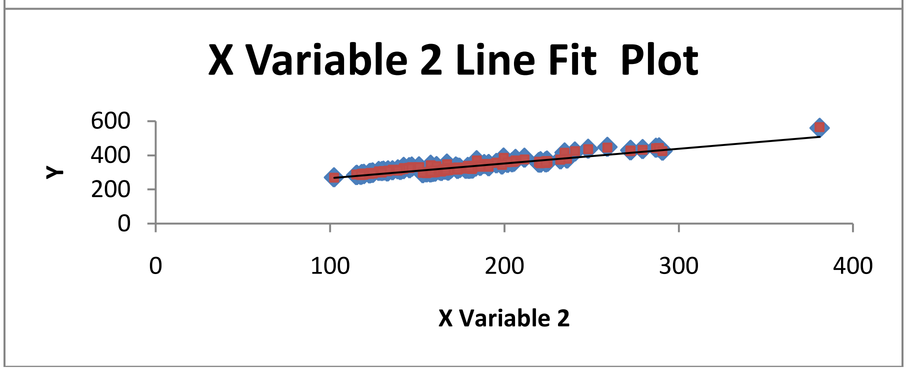
n37k5:


X Variable 1 Line Fit Plot
300
200
100
0
Y
3900
4000
4100
4200
4300
4400
X Variable 1

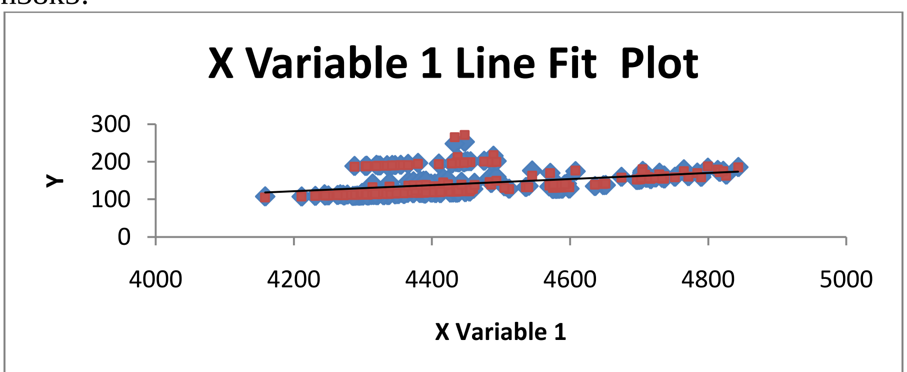

X Variable 2 Line Fit Plot
300
200
100
0
Y
0
20
40
60
80
100
X Variable 2

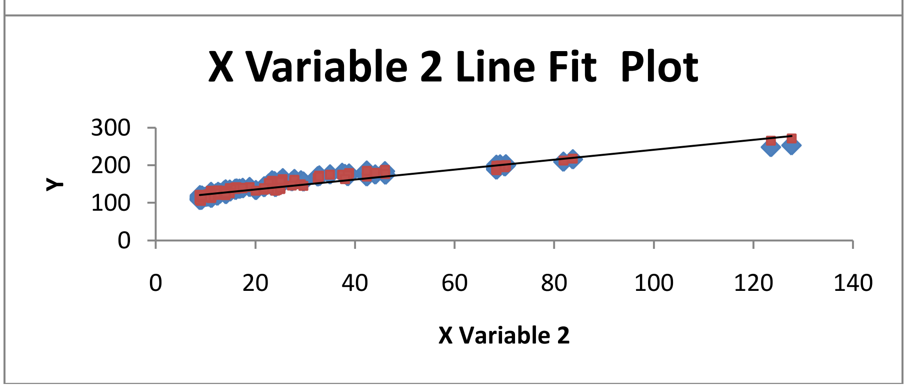
n37k6:


X Variable 1 Line Fit Plot
Y
600
400
200
0
0
1000
2000
3000
4000
5000
6000
X Variable 1

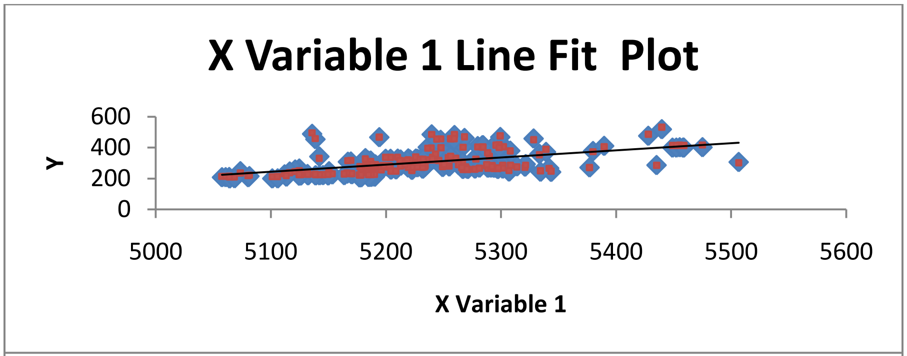

X Variable 2 Line Fit Plot
Y
600
400
200
0
0
100
200
300
400
X Variable 2

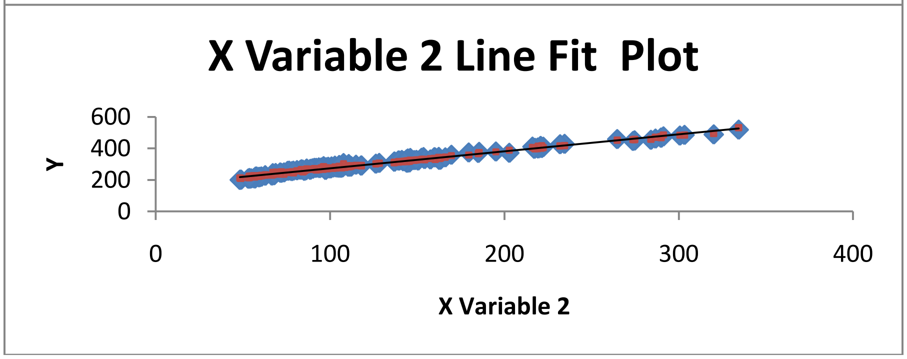
n38k5:


X Variable 1 Line Fit Plot
Y
300
200
100
0
4000
4200
4400
4600
4800
5000
X Variable 1

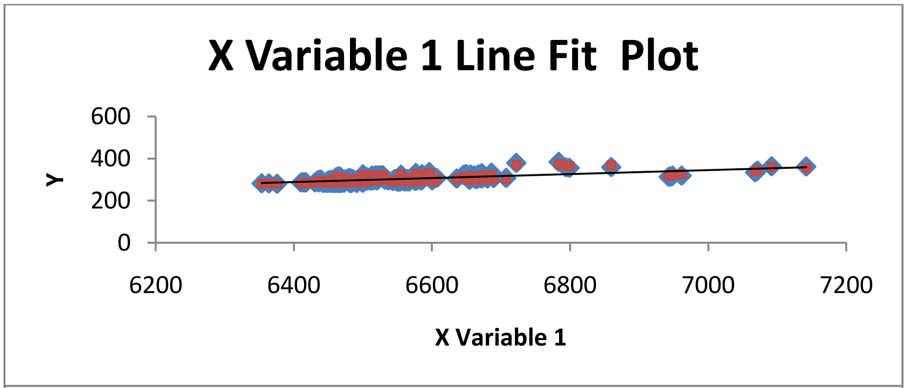

X Variable 2 Line Fit Plot
Y
300
200
100
0
0
20
40
60
80
100
120
140
X Variable 2

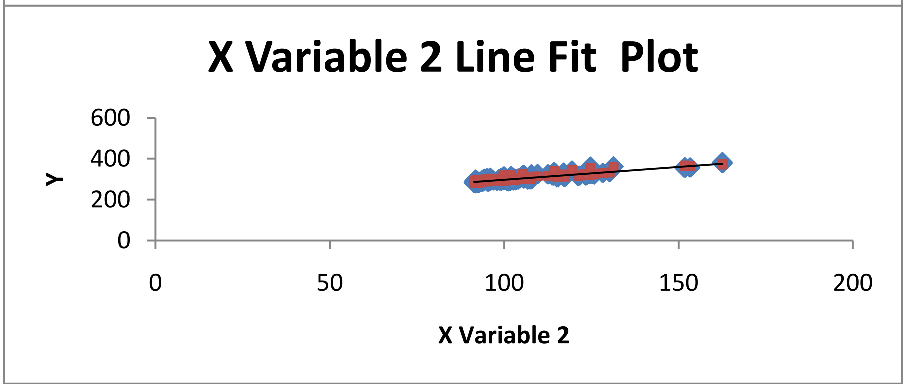
n39k5:


X Variable 1 Line Fit Plot
600
400
200
0
Y
5000
5100
5200
5300
5400
5500
5600
X Variable 1

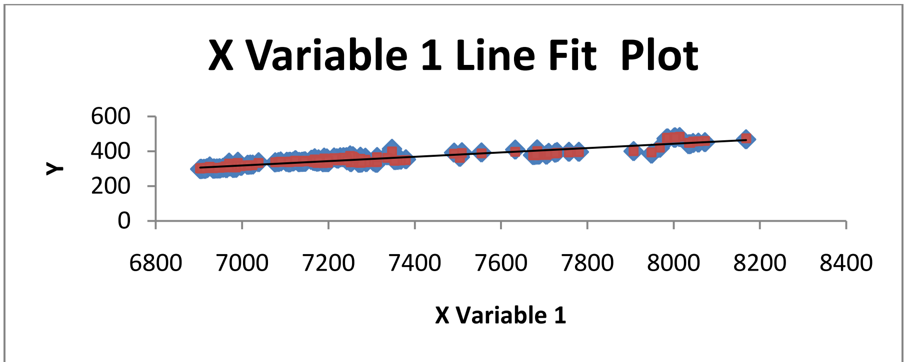

X Variable 2 Line Fit Plot
600
400
200
0
Y
0
100
200
300
400
X Variable 2

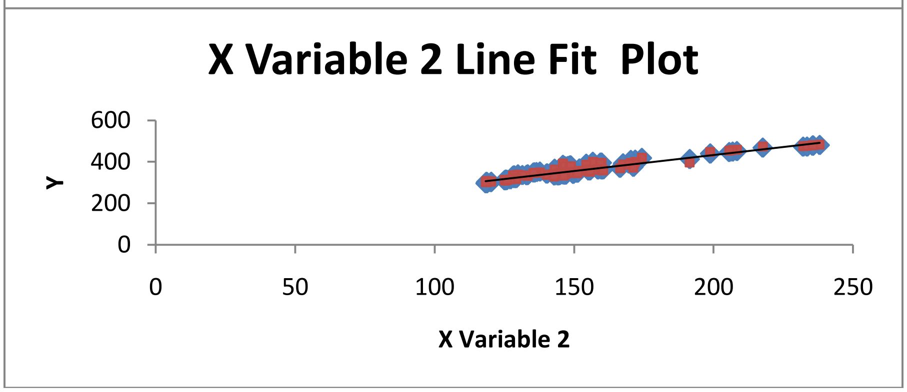
n53k7:


X Variable 1 Line Fit Plot
600
400
200
0
Y
6200
6400
6600
6800
7000
7200
X Variable 1

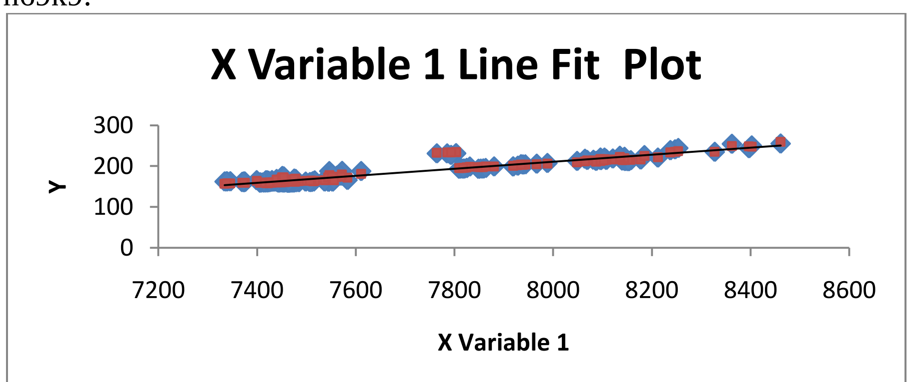

X Variable 2 Line Fit Plot
600
400
200
0
Y
0
50
100
150
200
X Variable 2

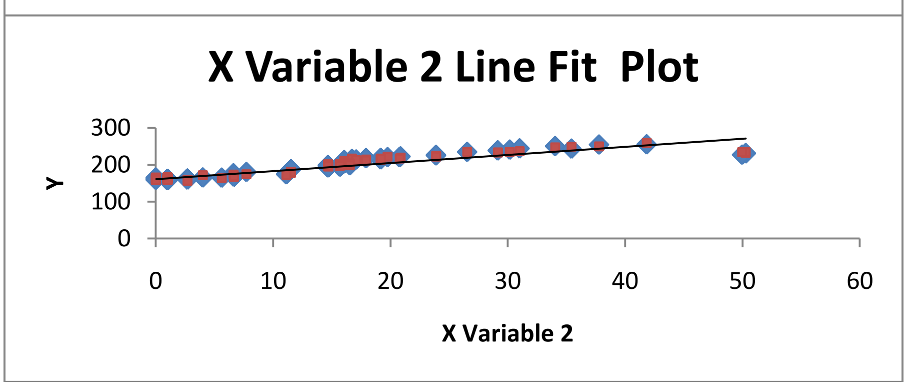
n60k9:


X Variable 1 Line Fit Plot
600
400
200
0
Y
6800
7000
7200
7400
7600
7800
8000
8200
8400
X Variable 1


X Variable 2 Line Fit Plot
600
400
200
0
Y
0
50
100
150
200
250
X Variable 2

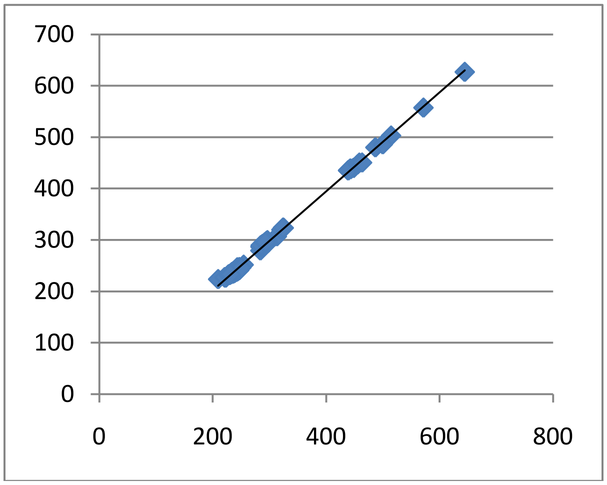
n69k9:


X Variable 1 Line Fit Plot
300
200
100
0
Y
7200
7400
7600
7800
8000
8200
8400
8600
X Variable 1

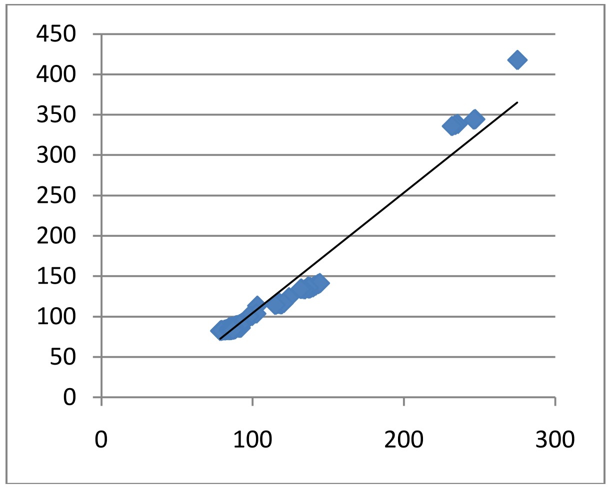

X Variable 2 Line Fit Plot
300
200
100
0
Y
0
10
20
30
40
50
60
X Variable 2

Model Line Fit Plots

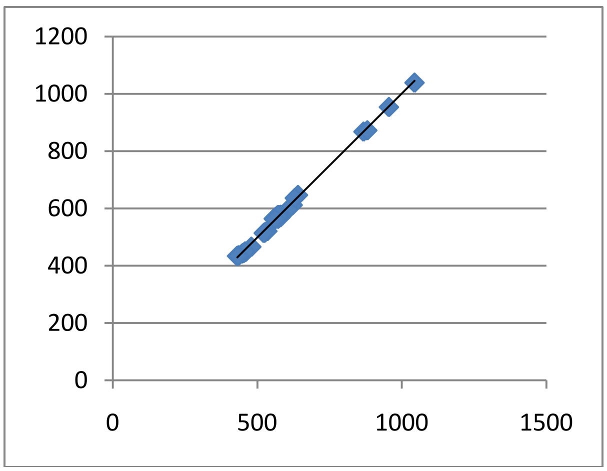
n32k5:



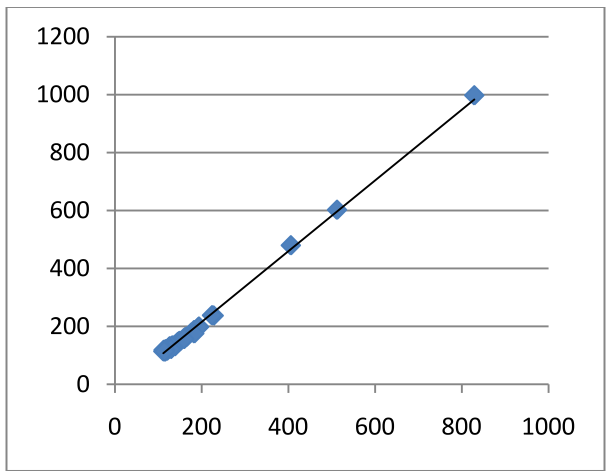
n33k5:



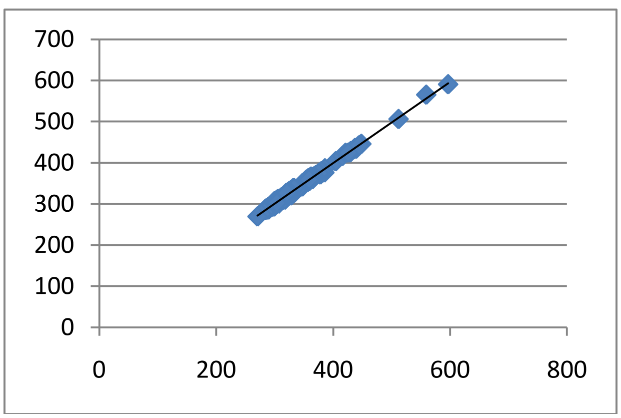
n36k5:



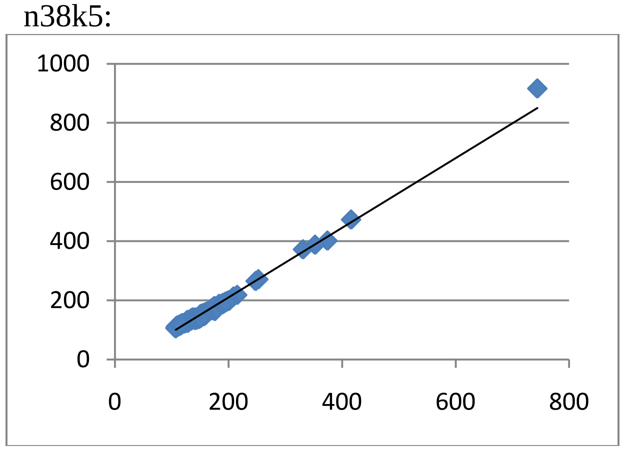
n37k5:



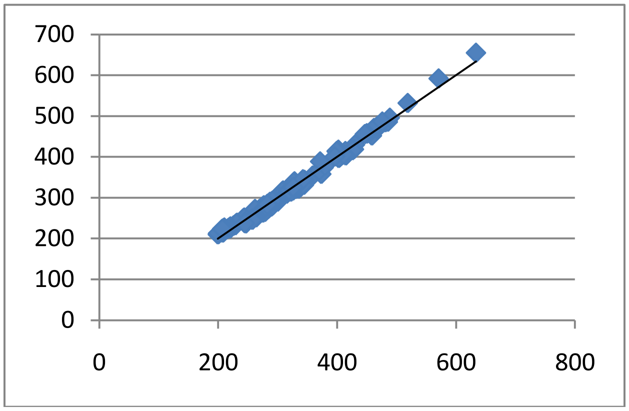
n37k6:



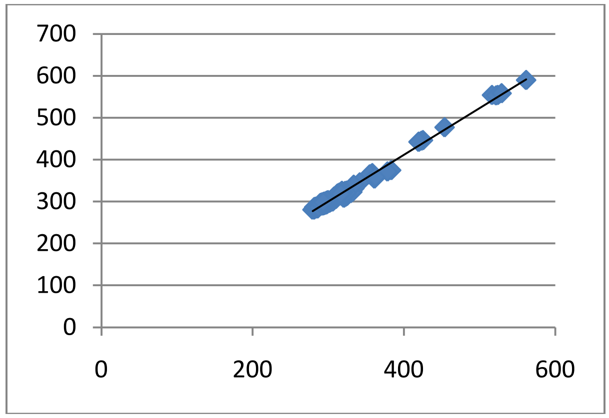
n38k5:



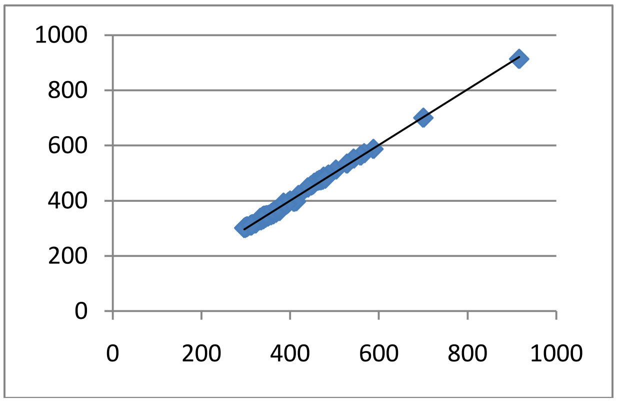
n39k5:



n53k7:



n60k9:



n69k9: