

# **An Application of VRP Algorithms with new Modifications**

**Melissa Restori**  
**Department of Industrial Engineering**  
**Arizona State University; Tempe, AZ 85281 USA**

## **Abstract**

This paper describes the algorithms behind a computer application that generates milk-runs for the retrieval of raw materials by the air-bag plant of TRW in Chihuahua, Mexico. In so doing, it presents different widely-used routing algorithms and discusses different modifications that were made to them both for practical and efficiency reasons. Formal testing and comparison of the algorithms and their modifications is left for future research. The entire project with TRW-Chihuahua for the improvement of its transportation operations, of which the computer application is part, has led to reductions in logistics costs of at least 30%.

## Table of Contents

|   |   |
|---|---|
| Section 1: Introduction.....                    | 1 |
| Section 2: Vehicle Routing Problem.....         | 1 |
| Section 2.1: .....                              | 2 |
| Section 2.2: Defining the problem.....          | 2 |
| Section 3: The Algorithms .....                 | 3 |
| Section 3.1: the Savings Algorithm.....         | 3 |
| Section 3.2: the Sweep Algorithm.....           | 4 |
| Section 3.3: Closest Insertion Algorithm .....  | 6 |
| Section 3.4: 2-opt exchange.....                | 7 |
| Section 3.5: Simulated Annealing .....          | 8 |
| Section 4: Conclusions and Future Research..... | 8 |
| Section 5: References.....                      | 9 |

## Table of Figures

|  |   |
|--|---|
| Figure 3.1: One-Way Savings heuristic .....                      | 4 |
| Figure 3.2: Sweep Algorithm.....                                 | 5 |
| Figure 3.3: Wedge Algorithm.....                                 | 6 |
| Figure 3.4: Closest Insertion Algorithm .....                    | 6 |
| Figure 3.5: 2-opt Exchange.....                                  | 7 |
| Figure 3.6: 2-opt Exchange with Closest Insertion Algorithm..... | 7 |

## **Section 1: Introduction**

Globalization has provided businesses with the ability to acquire and market products, resources and services throughout the world. This new business model is clearly different from the localized model of the past. As a result of globalization, companies have plants or subcontracts in countries that are thousands of miles apart, and have suppliers throughout the world. Consequently, the cost of the logistics necessary to receive raw materials, transfer parts between plants and deliver finished products to customers has sky-rocketed. It is apparent that as commercial needs broaden across the world, the need for transportation grows accordingly.

An example of globalization is given by the Maquiladora industry operating between Mexico and the United States. The defining characteristics of this industry are that most of the raw material originates in the US, the assembly is performed in Mexico and the final product is sold all over the world. This paper presents one component of a project that highlights the current logistics practices in the Maquiladora industry by examining the in-bound operations of a Maquiladora plant operating in Mexico. In particular, the focus is on the in-bound supply chain for the air-bag plant of TRW in Chihuahua, Mexico. TRW-Chihuahua's suppliers are located throughout the United States (though mostly in the Northeast) and southeastern Canada. TRW spent \$2.8 million in 2002 in the transportation of raw materials from its suppliers to its plant in Chihuahua.

During the analysis period of a project with TRW-Chihuahua, it was observed that the trucks that were arriving at TRW's in-bound consolidation center in El Paso, Texas were well below acceptable utilization levels. TRW-Chihuahua uses a third-party logistics company that designs the routes (or milk-runs) and contracts the trucking companies needed. Because of the poor truck utilization observed at the consolidation center, it was decided to try to develop a computer-based tool that would create the milk-runs needed to retrieve all of TRW-Chihuahua's raw material and compare the costs generated by the program to those charged by the third-party company.

The focus of this paper is to present and discuss the different algorithms that were used to solve the vehicle routing problem for the computer application. No attempt at this stage of the project has been made to formally rate the performance of the algorithms. Although most of the algorithms are well known, this paper presents modifications that were made either for applicability to this problem or for improved performance. Section 2 discusses the general vehicle routing problem and describes the specific one involved in the project with TRW-Chihuahua. Section 3 discusses the algorithms that were used and the modifications, if any, that were made to them. Section 4 presents the conclusions of the project and the future research that is still needed for a complete analysis of the algorithms. Section 5 lists the cited works of this paper.

## **Section 2: Vehicle Routing Problem**

This section begins by describing the general structure of a vehicle routing problem (VRP) in Section 2.1 and proceeds to describe the specific one that this paper is concerned with in Section 2.2.

## **Section 2.1:**

The traveling salesman problem (TSP) is a very common yet extremely complex problem. First developed in graph theory, it refers to the problem of finding the most efficient Hamiltonian circuit for a given set of points. A Hamiltonian circuit is a closed circuit that visits each node exactly once. The reason for the TSP's name is that this problem is similar to that of a traveling salesman who has to visit a certain number of clients in different cities and return home. He must visit each client exactly once. In this case, efficiency of the circuit is measured by its cost. Although this vehicle routing problem could be modeled using constraint programming, it is an NP-hard problem (Backer). This implies that there is no formulaic way of finding the optimal solution. Instead an exhaustive search of all possibilities may be the only way to find the optimal solution. Obviously, this approach is inefficient for large systems. For this reason practitioners turn to heuristics, which are algorithms that deliver "good" solutions but cannot be proven to be optimal.

Although the problem of developing a routing solution for TRW-Chihuahua's incoming raw material does not exactly fit the TSP model (the problem will be defined in Section 2.2), it is a VRP whose computational time for an exhaustive search of the possible combinations of the 150+ suppliers would be prohibitive. Thus, this project applies heuristics, which will be described in Section 3.

## **Section 2.2: Defining the problem**

As was mentioned in the previous section, the TRW-Chihuahua routing problem is a VRP though not exactly the TSP. Nevertheless, techniques that have been used by people trying to solve more traditional problems were employed and adjusted to fit the needs of TRW-Chihuahua's practical problem. This specific problem is one of assigning trucks to routes such that all of the ordered raw material available at the suppliers on a given week is brought to TRW-Chihuahua's consolidation center in El Paso, TX. The following is a list of assumptions that further shaped the problem:

- Truck trips are to be one-way. They will begin at the first supplier in the route and end at El Paso, TX.
- Cargo from one supplier will not be split between different trucks unless the cargo is greater than one truck-full. In that case, full trucks will be sent from the supplier directly to El Paso until less than one truck-full of cargo is left.
- The cost of the route is proportional to the distance traveled. Also, distances between all suppliers are assumed to be known and accurate.

Additional constraints to the problem are as follows:

- Trucks have dimension and weight restrictions (624"x101"x96" and 45,000 lbs, respectively).
- No more than five stops per route are allowed (that is, up to five suppliers can be visited by each truck, regardless of how much space is left over).
- Suppliers have set time windows (time frames where cargo is available for pick-up).
- Parts must be available at the consolidation center on a specific schedule.

As is obvious from this problem description, there are violations from the traditional TSP. There are two main divergences: instead of forming roundtrips (Hamiltonian cycles), the problem requires one-way routes and that instead of one route, the problem requires several. Also, different constraints have to be upheld.

## Section 3: The Algorithms

As in many application problems, similar problems have already been attempted and literature has been published about good ways to solve the problems. However, the specific nuances of a particular problem are rarely the same from one application to another. This is exactly the case with the TRW-Chihuahua routing application. Many algorithms are already well established and widely used, but searching for one that fit exactly the conditions outlined in Section 2.2 was fruitless. Modifications and improvements had to be made to those algorithms in order to work well for the problem.

Two different approaches to creating efficient routes were employed. The first of these, called “Cluster First and Route Second” involves first using algorithms to choose which cities should be included in a route and then using a different algorithm to optimally order those cities. The Savings Algorithm and the Sweep Algorithm are used for selecting cities while the Closest Insertion Algorithm is used to order the resulting clusters. Lastly, an improvement algorithm is applied, the 2-opt Exchange. The second approach employs the metaheuristic Simulated Annealing to create efficient routes without the need of improvement algorithms.

The following subsections present the algorithms used and the modifications that were made to them, if any were necessary. The clustering algorithms (Savings and Sweep Algorithms) are presented in Sections 3.1 and 3.2, respectively. The ordering algorithm (Closest Insertion Algorithm) is discussed in Section 3.3. The improvement algorithm (2-opt exchange) follows in Section 3.4. Finally, the Simulated Annealing metaheuristic is presented in Section 3.5.

### Section 3.1: the Savings Algorithm

Probably the best known heuristic for the VRP, the Savings Algorithm was developed by Clarke and Wright in 1964. As was mentioned in the previous section, this method (like the Sweep Algorithm, which will be discussed in the next section) is a clustering algorithm. This means that its purpose is to select the suppliers that will be included in a route and group them into a cluster. A different algorithm (presented in Section 3.3) will then set the order of the route. The Savings Algorithm finds pairs of suppliers that are beneficial in a route and links as many of the pairs as possible (Solomon). The steps for the heuristic are as follows (Clarke):

1. Begin with  $n$  dedicated routes (round trip), one for each of the  $n$  suppliers.
2. Compute savings in distance,  $S_{ij}$ , of combining every possible pair of suppliers<sup>1</sup>:  
$$S_{ij} = d_{0i} + d_{0j} - d_{ij}$$
3. In a list, order the savings in a decreasing fashion. Since negative  $S$  values are obviously undesirable, omit the negative values from the list.
4. Build a route by adding pairs that do not violate any of the set constraints (truck volume or weight, etc...) in the order they appear in the list until the route is full or the list has been exhausted.
5. Repeat Step 4 until all suppliers are routed or the list has been exhausted.
6. Any suppliers that were left unpaired (because of negative savings distances) are left as dedicated routes.

---

<sup>1</sup> Notation:  $d_{ij}$  is the distance from supplier  $i$  to supplier  $j$  and ‘0’ represents the consolidation center  
The formula comes from subtracting the new distances from the original ones and simplifying:

$$S_{ij} = 2*d_{0i} + 2*d_{0j} - (d_{0i} + d_{0j} + d_{ij}) = d_{0i} + d_{0j} - d_{ij}$$

The basic problem in applying this heuristic to the TRW-Chihuahua problem is that the heuristic assumes round trips. The heuristic was thus altered to create the One-Way Savings heuristic. The steps are as follows:

1. Find furthest unclustered supplier, F, from the consolidation center
2. Calculate One-Way Savings for all other suppliers not yet clustered:  

$$S_{ij} = d_{0F} - d_{Fi}$$
3. In a list, order the savings in a decreasing fashion. (Note that since F is the farthest supplier, S will never be negative).
4. Build a route by adding suppliers that do not violate any of the set constraints (truck volume or weight, etc...) in the order they appear on the list until the route is full.
5. Repeat Steps 1-4 until all suppliers are routed.

One way to picture what this heuristic accomplishes is to picture a line from the furthest supplier to the consolidation center. The heuristic tries to keep all suppliers in the route as close to that line as possible, thus aiming for fairly direct routes (see Figure 3.1).

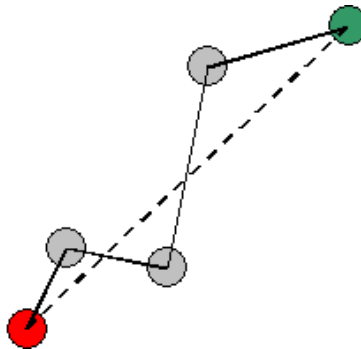


Figure 3.1: One-Way Savings heuristic

Although the One-Way Savings heuristic eliminates the necessity for round trips, it does have one shortcoming. Because the furthest supplier in each route is different, the list of savings calculation has to be recalculated every time that a new route is started (granted, with fewer and fewer suppliers at each iteration). For purposes of the TRW-Chihuahua project, however, the computational time for these added calculations was negligible in the application.

### Section 3.2: the Sweep Algorithm

The goal of the Sweep Algorithm is to obtain a cluster that is geographically close together. The terminology refers to assigning customers to the depot, which is essentially the same as assigning suppliers to a consolidation center. The steps for the algorithm are as follows (Gillett):

1. Locate the consolidation center and all suppliers on a graph, setting the coordinates such that the consolidation center is at the origin.
2. Start a radial sweep from the +x direction in a counterclockwise direction.
3. Assign all suppliers encompassed in the sweep to the cluster.
4. Stop when adding the next supplier would violate any of the constraints (vehicle weight or volume restrictions, etc).
5. Create a new cluster by start a new radial sweep where the last one left off.
6. Repeat steps 3-5 until all suppliers have been placed in a cluster.

The following figure (Figure 3.2) illustrates the Sweep Algorithm. The consolidation center is in red, the gray circles represent the suppliers in the first cluster, and the yellow circles are unclustered suppliers. The blue dotted lines and black arrow show the movement of the sweep in a counterclockwise direction.

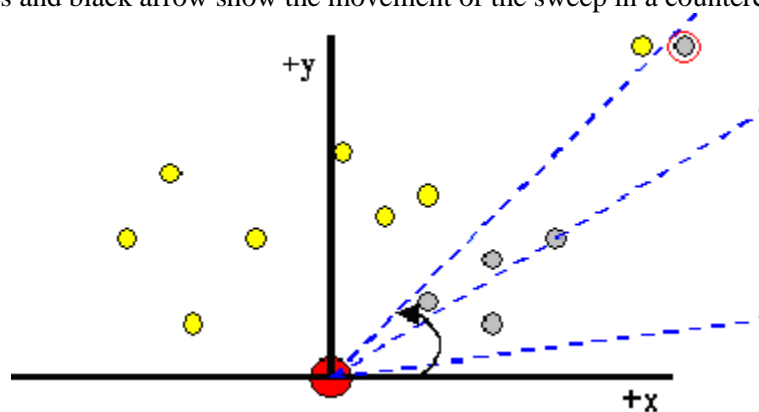


Figure 3.2: Sweep Algorithm

A basic flaw of the Sweep Algorithm is apparent in the figure (Figure 3.2). Consider the case (not drawn to scale) where the red circle represents El Paso, TX and the clustered supplier circled in red represents a city in the Northeast (Boston, MA for instance). The rest of the clustered suppliers would then be around Mississippi and Tennessee. Now notice that there is only one other supplier (unclustered) that is also in the Northeast. The problem that arises is that two different routes will have to begin in the Northeast even though there are only two suppliers there. Because of the specific set of data from TRW-Chihuahua, it was usually the case that few, very far removed cities in the East Coast would be split between different routes. For this reason, an alternative to the Sweep Algorithm was developed for the project, the Wedge Algorithm.

The Wedge Algorithm also bases its clustering using a radial sweep, but one that is more constrained than the one in the Sweep Algorithm. The steps for the Wedge Algorithm are as follows:

1. Choose the starting supplier that is farthest away from the consolidation center
2. Set an initial “wedge” of  $\theta$  degrees in either direction from the vector that connects the starting supplier and the consolidation center.
3. If clustering all the suppliers in the wedge does not violate any of the constraints, add them all and go to Step 4. Otherwise, order suppliers from closest to the vector to farthest (using straight-line distances). Add the cities to the cluster in that order until no more can be added without violating any constraints. Go to Step 5.
4. If the maximum allowed wedge angle,  $\beta$ , has not been reached, widen the wedge by a predetermined amount,  $\alpha$ , and repeat Step 3.
5. Repeat Steps 1-4 until all suppliers have been clustered.<sup>2</sup>

Figure 3.3 in the next page illustrates the Wedge algorithm. Again, the consolidation center is in red, the gray circles represent the suppliers in the first cluster, and the yellow circles are unclustered suppliers. The solid blue line represents the vector drawn from the consolidation center to the furthest supplier and the dotted blue lines are the limits to the wedge. As can be seen, only one route will have to travel the thousands of miles to the East Coast, thus greatly improving the clusters.

<sup>2</sup> Note: We used  $\theta = 5^\circ$ ,  $\beta = 10^\circ$  and  $\alpha = 40^\circ$ .

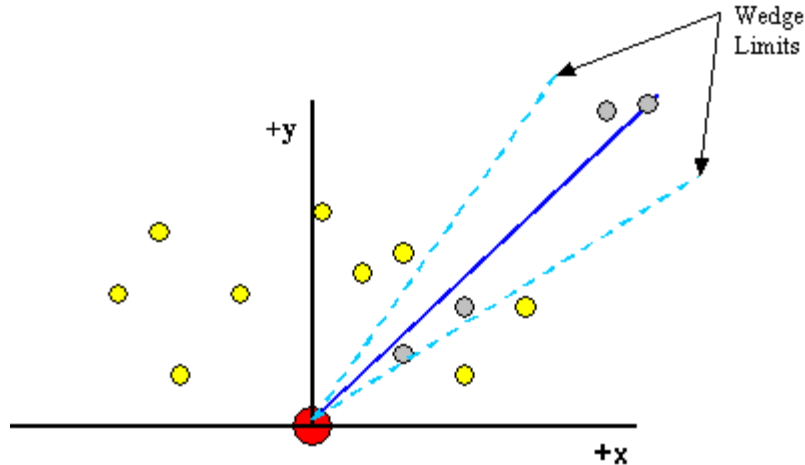


Figure 3.3: Wedge Algorithm

### Section 3.3: Closest Insertion Algorithm

As has been mentioned, the purpose of the Closest Insertion Algorithm is to order the clustered group of cities. Because of the constraints of the routes for the TRW-Chihuahua project, the arrangement that the Closest Insertion Algorithm develops is actually optimal for a given cluster. For this reason, the algorithm was incorporated without any changes. The steps for the algorithm are as follows:

1. Create a route with only the consolidation center in it.
2. Select any supplier from the cluster and add it to the route.
3. Calculate all the distances between unrouted and routed suppliers.
4. Choose the unrouted supplier that is associated to the minimum distance calculated in the previous step (break ties arbitrarily)
5. Calculate the total distances of the route that would result from inserting the chosen supplier at between all possible pairs of routed suppliers.<sup>3</sup>
6. Insert the chosen supplier where it yields the lowest of the distances calculated in the previous step.
7. Repeat steps 3-6 until all clustered suppliers have been routed.

The left side of Figure 3.4 shows the route with suppliers i, j, k and l already placed. Supplier u is the supplier being added to the route. The distance generated by placing supplier u between i and j are calculated, as well as between all other pairs of routed suppliers. Finally, as is seen in the right side of the arrow, it places u between j and k, its optimal location.

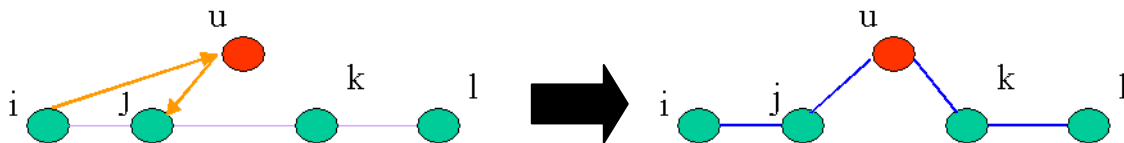


Figure 3.4: Closest Insertion Algorithm

<sup>3</sup> For one way routes (instead of round trips), do not allow any supplier to be placed after the consolidation center, which will be final destination.



### Section 3.4: 2-opt exchange

The 2-opt exchange is a very simple, yet very useful, improvement heuristic. It involves exhaustively considering exchanges of two suppliers in different routes; for a given exchange, if the resulting mileage is lower than before the exchange (and no constraints are violated), the exchange takes place (Lin). The process is repeated until no more exchanges produce a decrease in mileage. Figure 3.5 illustrates what two routes would look like after undergoing a 2-opt exchange.

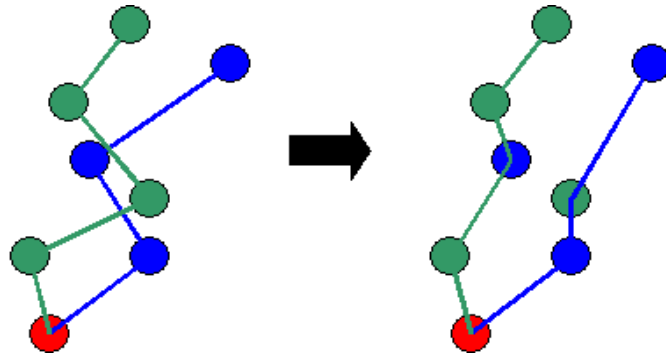


Figure 3.5: 2-opt Exchange

The 2-opt exchange is actually just one case of a broader heuristic, the  $k$ -opt exchange, where  $k$  is the number of suppliers considered for exchange at each iteration. Thus a 3-opt exchange is where three suppliers are exchanged to test for reduced mileage. Obviously, as  $k$  increases, the solution improves, but at the cost of an exponentially greater number of iterations (in general, the  $k$ -opt exchange has complexity of  $O(N^k)$ ). For this reason, only the 2-opt exchange was applied to the TRW-Chihuahua project.

The only modification made to this algorithm is applying the Closest Insertion Algorithm to the two routes affected by a possible exchange before deciding whether to keep the altered routes or not. The purpose of this added step is because when an exchange is considered, the resulting routes may not be in their optimal arrangement. Figure 3.6 helps illustrate this case. The middle figure shows the new route as generated by exchanging the two cities without the Closest Insertion Algorithm. Obviously the route is not in its optimal order. Performing the Closest Insertion Algorithm, as in the last step in the figure, ensures that only optimally ordered routes are compared in mileage.

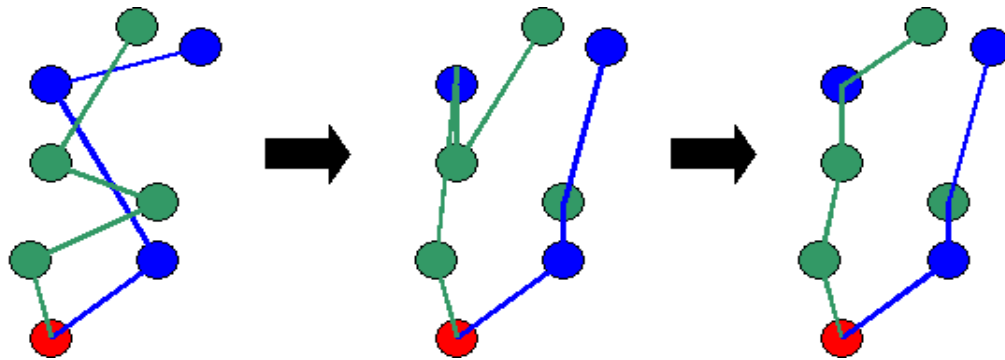


Figure 3.6: 2-opt Exchange with Closest Insertion Algorithm

A somewhat related heuristic developed for the application was to consider instead of exchanging suppliers, simply moving a supplier from one route to another. This was named the balancing heuristic for two reasons; first, it tends to add suppliers to routes that are still below capacity and second, it tends to eliminate routes that only have one stop in them.

### **Section 3.5: Simulated Annealing**

A problem that is inherent in the previous heuristics is that they become trapped within a local optimum (Lau). Consider for example the 2-opt exchange. It does not consider future moves at all, only the current two suppliers to be exchanged. It many times is the case that making one exchange now that slightly increases the mileage will allow for an exchange in the future that will dramatically decrease the mileage. However, the intermediate “worsening” step is not permitted in the 2-opt Exchange and thus the desirable exchange will never occur. This is what is meant by being trapped in a local optimum and it is a “significant factor affecting the quality of solutions” (Li).

Several metaheuristics have been developed to escape local optima traps. The one that was chosen for use was simulated annealing, which is based on the “analogy to the way that metals cool and anneal as temperatures change” (Sohn). At high temperatures (in the molten state) atoms have high energy and can move freely throughout the metal. If the metal were to be instantaneously cooled to a solid state, the atoms would virtually freeze in place instead of moving to the lowest-energy crystal arrangement. On the other hand, slow-cooling (annealing) the metal would allow the atoms to have enough thermal energy during the solidification stage to move to their optimal location. Similarly, by imposing a probability distribution on the routes according to their total distance, it is possible to loosen the constraint of distance minimization enough to allow “escapes” from local optima. This freedom to explore “worse” routes is analogous to the effect of thermal energy in a metal. The following paragraph describes the actual steps in Simulated Annealing.

Routes are first created randomly. Then steps similar to the 2-opt exchange occur, only that for an exchange to be accepted, the total mileage need not decrease. Instead, a certain probability is given to accepting a mileage increase (the probability is an exponentially decreasing function of the distance by which the route worsened with the exchange). With time (measured by number of iterations), the probability is slowly reduced until accepting an exchange that would worsen the route is zero (the reduction in probability is analogous to the cooling rate of the metal). Obviously, with an infinite number of iterations at each probability level and infinitesimally small “cooling rate,” the metaheuristic would yield the optimal solution (since all combinations would be attempted). Practically, however, it is an “efficient approach for solving difficult combinatorial problems (e.g. traveling salesman problem)” (Chiang). Its drawbacks are finding good values for the parameters (initial probability level, cooling rate and number of iterations per probability level) and the computational time required for good results.

## **Section 4: Conclusions and Future Research**

As was mentioned in the introduction, this paper does not provide data on which algorithm(s) is best or under what circumstances each should be used. This study would be of interest to the author and is left as a future extension to the project. The goal of the paper is to describe the heuristics used for the application with TRW-Chihuahua and describe any alterations that were made to those heuristics.

The final design for the computer application involves using all of the algorithms discussed in three different route-generating methods. The first and second methods use the adaptations of the Savings and Wedge Algorithms, respectively, for clustering followed by the Closest Insertion Algorithm for ordering

and are then improved by the 2-opt exchange and balancing heuristics. The last method uses only Simulated Annealing. The output of the program is the route of the algorithm that yielded the lowest mileage. In general, when given sufficient iterations, Simulated Annealing outperforms the rest of the algorithms. There is no noticeable trend in the performance of the two other methods.

The focus of the project with TRW-Chihuahua was to reduce the transportation costs of the plant. The computer application that was developed served as an initial comparison against the performance of its third-party logistics company. Thanks to the project, TRW-Chihuahua was able to assess that the logistic company was not efficiently routing the raw material. As a result, TRW-Chihuahua has changed its third-party company and has reported savings of over 30%.

## Section 5: References

Backer BD, V. Furnon, P. Shaw, P. Kilby, P. Prosser. "Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristic." *Journal of Heuristics*. 6 (4): 501-523 (SEP 2000).

Chiang, W.-C. and R.A. Russel. "Metaheuristics for the vehicle routing problem." *Annals of Operations Research*. 63: 3-27 (1996).

Clarke, G. and G.W. Wright. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." *Operations Research*. 12: 568-581 (1964).

Gillett, BE and Leland R. Miller. "A Heuristic Algorithm for the Vehicle-Dispatch Problem." *Operations Research*. 22: 340-349 (1974).

Lau HC, Sim M, Teo KM. "Vehicle routing problem with time windows and a limited number of vehicles." *European Journal of Operational Research*. 148 (3): 559-569 Aug 1 2003

Li, Haibing, Andrew Lim. "Local search with annealing-like restarts to solve the VRPTW." *European Journal of Operational Research*. 150 (003) 115-127

Lin, S. "Computer Solutions of the Traveling Salesman Problem." *Bell Systems Technology Journal*. 44: 2245-2269 (1965).

Sohn, Andrew. "Generalized speculative computation of parallel simulated annealing." *Annals of Operations Research*. 63: 29-55 (1996).

Solomon, Marius M. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research*. Vol. 35, No. 2, March-April 1987