A Tabu Search Algorithm for the Safe Transportation of Hazardous Materials

L. Zhang, S. Guo
Department of CS
Zhongshan University
135 Xingangxi Road
Guangzhou, P.R.China
is00zlei@student.zsu.edu.cn
isdcs05@zsu.edu.cn

Y. Zhu
Department of CSE
Univ of California, San Diego
9500 Gilman Drive
La Jolla, CA, the United States
y2zhu@cs.ucsd.edu

A. Lim
Department of IEEM
Hong Kong Univ of Sci & Tech
Clear Water Bay
Kowloon, Hong Kong
iealim@ust.hk

ABSTRACT

In this work, we study the problem of the safe transportation of hazardous materials, which is an important operational problem and has been studied extensively in the literature. We outline previous work and propose a new model that is a variant of the vehicle routing problem with time windows (VRPTW). The objective is to find a schedule to guarantee the safety of all vehicles. We propose a tabu search (TS) heuristic with a dynamic penalty mechanism to obtain good solutions. A realistic data generation mechanism is also presented and the elaborate computational results show the strengths of our algorithms.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: [Sequencing and scheduling]

Keywords

Heuristics, Optimization, Tabu Search, Vehicle Routing

1. INTRODUCTION

The transportation of hazardous materials (hazmat) is a significant issue in the transportation industry. There are billions of tons of hazmat transported throughout the world every year, most of which are important raw materials used by various industries. During the transportation process, one of the most important issues is safety. Accidents, though rarely happening, could have large impacts on humans and the natural environment, especially in large urban areas [14]. In 1999, the U.S. Department of Transportation reported 5 fatalities, 217 injured persons, and $23 million in property damage associated with the transportation of hazardous materials on highways [10].

As an important strategic and tactical problem, the transportation of hazmat has been studied extensively in the research literature. In the early 1990s, the problem was highlighted by the researchers and modelled to achieve different objectives, including the minimization of risk, the minimization of risk to special population categories, the minimization of travelling time, and the minimization of property damage[13]. According to List et al. [9], research in this field includes risk analysis, routing/scheduling and facility location. Karkazis and Boffey [7] introduce an improved routing model in the context of a realistic environment and solve the problem by a branch-and-bound algorithm. Erkut and Ingolfsson [2] suggest that avoiding a catastrophe (an accident with a very large consequence) is a relevant issue in routing hazardous materials and propose three models, two of which are computationally hard and therefore only offer theoretical insights. Leonelli et al. [8] employ a risk-analysis-based routing methodology to solve the hazmat transportation problem and formulate it as a minimum cost flow problem. A survey in 2002 by Luedtke [10] summarizes some research results on hazmat transportation. Zografos and Androutsopoulos [12] present a new model that is based on a bi-objective vehicle routing and scheduling problem. A heuristic is proposed to solve the new model and it is applied to several benchmark problems.

In the other direction, Erkut and Verter [4] analyze the different prevailing risk models using the US road network and find that they usually provide different "optimal" solutions, therefore researchers and practitioners are advised to choose the models carefully in hazmat transportation. Two axioms are mentioned by Erkut and Ingolfsson [3] to check the qualities of the previous risk models. There is also a unique paper by Kara and Verter [6] focusing on the nature of the relationship between the regulator and carriers. It is a government's authority to close certain roads to hazmat transportation, while it is carrier's strategy to choose roads to route their vehicles. This paper provides us a new perspective to study the field.

The vehicle routing problem with time windows (VRPTW) is a well-known operational research problem in transportation studies [11] and has been well studied during the past several decades. It is interesting to study the modelling and solution approaches when hazmat are transported by vehicles, and the solutions are useful in the real-world applications. However, in the extensive literature, only the recent papers [12] have considered the relationship between the transportation of hazmat and VRPTW. In this new model, the authors suggest that each road segment, (i,j), has a travelling time, c_{ij}, and a risk, R_{ij}. The bi-objective function is to minimize the total travelling time, z_1, and the total risk, z_2. However, when using heuristics to solve the model, the bi-objective function is converted to a single objective function, $z = w_1 z_1 + w_2 z_2$, which is a weighted sum. In fact, we find that it is equivalent to the traditional VRPTW problem if we modify the travelling time of the road segments, (i,j), to $c'_{i,j} = w_1 c_{ij} + w_2 R_{ij}$. Therefore, we can apply the available methods for the traditional VRPTW to the new model directly and the heuristic proposed in [12] appears not so useful.

We also note there is a weakness in this model. The volumes of hazmat transported are omitted when considering the risk of a specific road segment. Therefore, the risk of a fully loaded vehicle and an empty vehicle is the same. To our knowledge, however, the risk in transportation is highly related to the volume of hazmat in the vehicles. Only when the hazmat in the environment is over a certain level will it threaten the people and other living creatures nearby. We should therefore consider that the transportation is safe if the load of the vehicles is under a certain "safety level".

The safety levels are different in various environments. For example, in the areas with low people densities (such as a desert), vehicles are allowed to carry a relatively larger volume of goods as the impact a leak would not be very harmful; on the contrary, if vehicles pass through a large city, a small leak will result in a disaster.

Based on the above-mentioned analysis, we propose a new model of safe transportation of hazardous materials based on VRPTW. Each road segment, (i,j), is associated with a value, r_{ij}, which is the upper bound of the load for vehicles passing along this road segment and determined by the nearby environment. Therefore, the complete scheduled transportation route is safe if the loads of all the vehicles do not exceed the upper bound of each road segment when passing along it. Figure 1 illustrates a clear example.

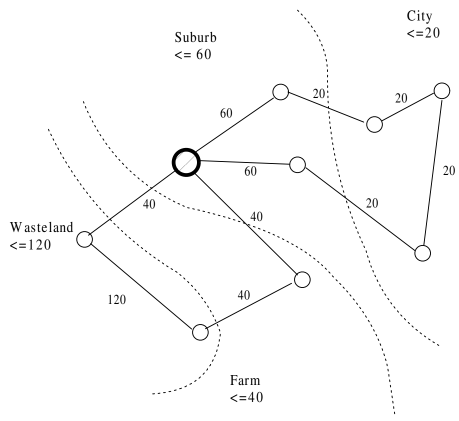

Figure 1: An Example for the New Model of Hazardous Transportation

The paper is organized as follows: we give the definition of the new model in the next section; a Tabu Search algorithm is given in section 3; section 4 shows the computational results of our algorithms on the benchmarks; the conclusion is presented in the last section.

2. PROBLEM DEFINITION

The safe transportation problem of hazmat is a variant of VRPTW, which considers the vehicle scheduling problem between a depot (a distribution center of hazmat) and several customers. There is one and only one road segment between the depot and each customer or each pair of customers. Each road segment has an upper bound for transportation quantity. The transportation is regarded as "safe" if the load does not exceed the upper bound when a vehicle is passing through a specific road segment. The objective is to minimize the number of vehicles used and the total travelling distance, provided that the transportation used is safe on all road segments.

There are several constraints inherited from VRPTW, in addition to the above-mentioned ones:

1. Each vehicle has its own path, which starts from the depot, goes to one or more customers, and terminates at the depot.

2. Each customer has to be served once and only once. All customers' demands must be satisfied.

3. Each customer has a time window $[a, b]$; the hazmat must be delivered within this time window. The vehicle has to wait if it arrives before time a. Each customer has a delivery time (service time); the vehicle cannot leave the customer until the delivery is complete.

4. The load of vehicles cannot exceed the capacity.

The following notations are used throughout the paper:
n: number of customers;
K: number of vehicles;
a_i: lower bound of customer i's time window;
b_i: upper bound of customer i's time window;
r_{ij}: the maximum quantity allowed along the road between node i and node j;
w_{ik}: the moment that vehicle k arrives at node i;
u_i: the load of vehicle before it arrives at node i;

The integer program model is not presented here due to the space limitations.

3. A TABU SEARCH HEURISTIC

As the problem is NP-Hard (if we set the upper bound on each road segment to be infinity, the problem is transformed to the traditional VRPTW), we cannot propose a polynomial time algorithm unless P=NP. Therefore, an efficient heuristic algorithm is useful in practice. Here, we develop a tabu search algorithm to obtain good quality solutions. Tabu search (TS) is a meta-heuristic search procedure that proceeds iteratively from one solution to another by moves in a neighborhood space with the assistance of adaptive memory [5].

The different components of the TS algorithm, such as the initial solution generation, the dynamic penalty mechanism, and neighborhood search operators are discussed in the remainder of this section.

3.1 Initial Solution Generation

The entire process of TS is to iteratively improve the initial solution. To capture more information, we allow our initial solutions to be infeasible, i.e., the initial solutions generated might violate some constraints. However, the degree of "infeasibility" must be controlled enough to avoid negative impacts. For example, if we allow for both the time window and the road segment upper bounds constraints to be relaxed, such a solution may occur as there is only one path that passes through all customers. Of course, this solution is totally meaningless.

Therefore, we use the following algorithm to generate initial solutions (R represents the current set of paths; a path $r = (0, c_1, c_2, \ldots, c_k, n+1)$):

1. $R = \emptyset$;
2. If all customers are serviced, go to (5).
3. For each customer, c, that has not been serviced, for all $r \in R$, try to insert c into r (try all positions). A most suitable insertion position is selected, subject to:
   (a) no constraints are violated because of the insertion;
   (b) the total travelling distance is minimized after the insertion.
4. (a) If such a position exists for customer c, then c is inserted and go to (2);
   (b) otherwise randomly choose a customer, c, construct a new path $r' = (0, c, n+1)$, $R \leftarrow R \cup \{r\}$; go to (2).
5. End.

Note that infeasible solutions may be produced in step 4(b). Actually, in the iterative improvement stage of the TS process, these infeasible paths will be eliminated by being inserted in other paths. In addition, in Step 3(a), we have to impose the constraints, otherwise the above-mentioned "solution" with only one path would occur, which is not what we expect.

3.2 The Dynamic Penalty Mechanism

Neighborhood search is a crucial step in the TS process. It is this step that determines the quality of the solution. Generally we favor the solutions with smaller objective values. When generating the neighborhood solutions, however, we allow infeasible solutions. The two reasons are stated as follows: firstly, if the initial solution is infeasible, we cannot expect that a feasible solution could be obtained in one or two iterations; secondly, and more importantly, it is easy to trap in the local optima if we only focus on eliminating the "infeasible components" of solutions, or do not even permit any infeasible solutions. In fact, the search without permitting infeasible solutions performs poorly in the experiments.

The two objectives — permitting infeasible solutions and searching for better solutions — seem contradictory to each other. Therefore, we introduce a new scheme called "dynamic penalty" as a compromise. A similar idea is mentioned in [1] but it is not the same as what we suggest here.

As we can observe, there are mainly two situations when a road segment is "infeasible" for a vehicle: 1. the vehicle arrives after the time window; 2. the load of the vehicle exceeds the upper bound of the road segment. Therefore, we should penalize the infeasible solutions if either of them is encountered. Let $S_1 = \{(i,k)|$ if vehicle k violates the time window constraint at customer $i\}$, $S_2 = \{(i,j)|$ if the upper bound load constraint is violated at segment $(i,j)\}$, The penalty function is calculated as follows:

$$Penalty = P_l \sum_{(i,k)\in S_1} (w_{ik} - b_i) + P_o \sum_{(i,j)\in S_2} (u_j - r_{ij}) \quad (1)$$

The definitions of the variables w, b, u, r have been given in Section 2. P_l and P_o are the late penalty coefficient and the overload penalty coefficient, respectively, which change dynamically during the search process. If in a series of iterations the solutions obtained are always infeasible, the coefficients are multiplied by 2 to penalize infeasible solutions and to guide the algorithm to find a feasible solution; on the other hand, if the solutions remain feasible for consecutive iterations, we could halve the coefficients to relax the constraints and expect a jump from the local optima.

3.3 Penalty Regression

The allowance of infeasible solutions and the introduction to the dynamic penalty mechanism enable the TS algorithm to explore the search space more efficiently. Sometimes, however, a problem may appear and impair the search ability. If we consecutively choose feasible solutions, the values of P_l and P_o will become very small; after that, infeasible solutions are selected in the following iterations. Although the values of P_l and P_o can be increased according to the aforementioned rules, the effects will not be as good as expected as they are increased from small values. The solutions obtained may therefore violate too many constraints and be difficult to recover.

We use the following "penalty regression" scheme to remedy this problem. When P_l or P_o has exceeded an upper bound, which means infeasible solutions have appeared in a series of consecutive iterations, the most recent feasible solution is selected as the current solution, the values of P_l and P_o are kept and the search process is continued. Actually, this scheme introduces the idea of "backtracking" to the TS process, which eliminates some negative impacts of the dynamic penalty scheme.

3.4 Move Evaluation

TS is an aggressive heuristic that always chooses the most valuable neighborhood solutions. In this problem, we do not treat the solution with the least objective value as the most valuable solution. A valuable solution should be a solution that has a potential to improve, regardless of its feasibility. Therefore, the following evaluation function is applied in our algorithm.

Let K' denote the number of infeasible paths and let L represent the total travelling distance. The evaluation function is

$$eval = L + Penalty + \alpha(K'/K)L + \beta(K/n)L \quad (2)$$

where the definition of $Penalty$ has been given above and α and β are constants.

The third term lowers the values of infeasible paths, while the fourth term emphasizes the number of vehicles during the evaluation. These two terms help the search process.

The detailed experimental results are presented in the next section.

3.5 Neighborhood Search Operators

We apply four neighborhood search operators here: single point insertion, tail exchange, path decomposition and new path creation.

3.5.1 Single Point Insertion

In the current solution, we select a customer, c, and insert this customer into another path r'. If we try each customer, the complexity for this operation would become $O(n^3)$, as we have to check the feasibility of r' after insertion. The complexity is high for a single neighborhood operation (consider the normal scale of $n = 100$). We therefore select the following "potential customers" for whom adjustments are possible.

- The time window constraint or the load constraint is violated when a vehicle passing through a customer c, where no constraints have been violated before. The reason could be that the vehicle should not choose the road to get to c. Therefore, we remove the customer c from the current path and insert it to another path;
- After a vehicle passes over a road segment (c_i, c_j) with a small load upper bound, its ability to service the succeeding customers along the path will be greatly reduced. It may be "wrong" to schedule the vehicle to pass over this road segment. Therefore, we try to insert customer c_j to another path;
- When the time of a vehicle arriving at customer c is much later than the starting time of the pre-determined time window of c, it is easy for the vehicle to violate the time window constraints of the succeeding customers along the path. It might be better to avoid this customer c and insert it into another path.

We have tried to perform single point insertion for all customers and for potential customers. The results indicate not much difference. Therefore, it is possible to choose potential customers to reduce the complexity of the neighborhood search without lowering the exploration ability.

3.5.2 Tail Exchange

In order to diversify the neighborhood space and to avoid the local optima, we introduce the tail exchange operator. Due to the complexity consideration, we can only try some "potential" exchanges instead of all the combinations. In the implementation, for two vehicles, k_1 and k_2, with the paths r_1=(0, …, $c_{g_{i-1}}$, c_{g_i}, $c_{g_{i+1}}$, …, $n+1$) and r_2=(0, …, $c_{g'_{j-1}}$, $c_{g'_j}$, $c_{g'_{j+1}}$, …, $n+1$), respectively, we want to perform the exchange of the two tails starting from c_{g_i} and $c_{g'_j}$, if the following conditions are satisfied:

- The distance between c_{g_i} and $c_{g'_j}$ is within a certain range;
- The arrival time for vehicle k_1 to c_{g_i} and the arrival time for vehicle k_2 to $c_{g'_j}$ are close;
- The servicing abilities (load after servicing the current customer) for vehicle k_1 after customer c_{g_i} is close to the total demand of the customers in r_2 after customer $c_{g'_j}$; the similar to vehicle k_2;

After the exchange, the path for k_1 is r_1=(0, …, $c_{g_{i-1}}$, $c_{g'_j}$, $c_{g'_{j+1}}$, …, $n+1$) and the path for k_2 is r_2=(0, …, $c_{g'_{j-1}}$, c_{g_i}, $c_{g_{i+1}}$, …,$n+1$).

3.5.3 Path Decomposition

This operator is extremely useful to reduce the number of vehicles. Although single point insertion can also reduce a path (when the path has only one customer), this situation rarely occurs. The insertion operator is therefore mainly used for small adjustments. However, the path decomposition operator, which decomposes an entire path and inserts the customers into other paths, is more "powerful" than the single point insertion operation in eliminating paths.

Assume that the current set of paths is R={r_1, …, r_{i-1}, r_i, r_{i+1}, …, r_k}. If we decompose path r_i=(0, c_{g_1}, …, c_{g_m}, $n+1$), the algorithm runs as follows:

1. $R'_0 \leftarrow R - \{r_i\}$;
2. $j \leftarrow 1$;
3. if $j > m$, goto (7);
4. for each customer c_{g_j} in r_i, try to insert the customer into every position of each path in R'_j, calculate the incremental value of the evaluation function, choose the position with the least incremental value;
5. insert c_{g_j} to this position and get a new set R'_j;
6. $j \leftarrow j + 1$, goto (3);
7. take R'_m as the neighborhood solution.

3.5.4 New Path Creation

This operator is used to remove some customers from their original positions and insert them to other positions that may be more "appropriate". Of course, the insertion operator can also do this, but with more iterations. There is another way to represent the current solution as a pair (R, C), where R is the current set of paths and C is the set of customers that are removed from the paths and to be inserted. The value of the current solution is the sum of the evaluation function for R and the penalty for C. Each time we generate the neighborhoods, we choose some (or all) customers from C and insert them to R, or remove some customers from R or put them in C.

Note that the last two terms of the evaluation function penalize the infeasible paths. By taking the advantage of this, we can simplify the above operations by constructing a new path for the customers in C and add it to R. The penalty for C is automatically taken care of by the evaluation functions.

On the implementation of the new path creation, one thing we should emphasize is that we remove two customers from the current solution instead of a single customer every time. Otherwise, it is possible that the removed customer will be inserted to other paths in the next iteration, which creates the same effect as the single point insertion operation, but with one more iterations.

This operator also shows a good performance along side the path decomposition operator. Figure 2 gives an example.

3.6 Tabu Memory

For the single point insertion operation, if a customer, c, leaves a path, r, and is inserted to another path, r', it is

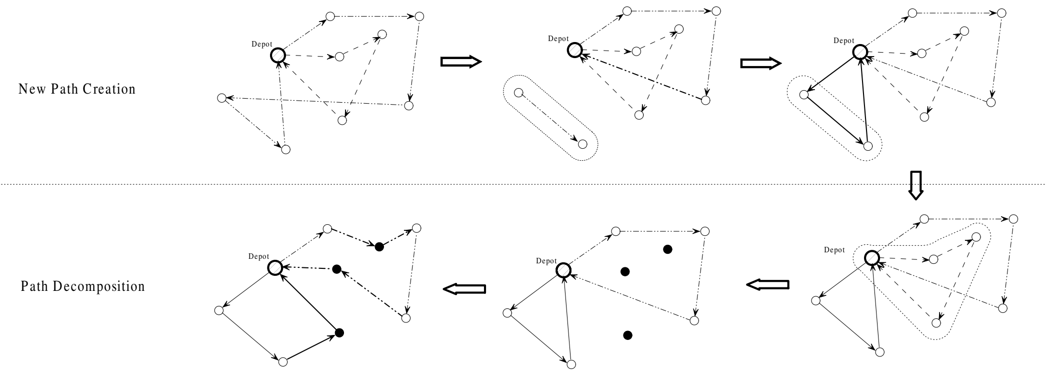

Figure 2: An example for the corporation of Path Decomposition and New Path Creation

forbidden to return the customer to the original path within a certain number of iterations. The pair (r, c) is recorded in the tabu memory. The pair (r, r') is recorded in the tabu memory to prevent another tail exchange operation between r and r' after a tail exchange operation between the paths r and r'. We forbid the entire paths r and r' because this operation is not expected to occur frequently. As the reverse operation for the path decomposition is highly impossible, we do not impose any restrictions. For the new path creation operation, the pair (r, c) is recorded in the tabu memory, where r is the first path from which customers are removed and c is the first customer that is removed from the current solution.

The *tabu tenure*, which determines the active duration of the tabu memory, is also an important parameter in TS and affects the solutions greatly. After the extensive experiments, we find that $4n - 5n$ is a suitable range for the *tabu tenure.*

4. EXPERIMENTAL RESULTS

As the model is new and no test instances are available, we cannot measure the performance of our algorithm by comparing it with previous results. Therefore, we create our own benchmarks by modifying Solomon's VRPTW instances [11] and compare the performance of different variants of TS. We also modify our algorithm to run with the standard VRPTW benchmarks. All the algorithms are implemented using C++ and run on a Pentium IV 2.0G, 512M RAM PC. The detailed data generation mechanism and computational results are presented below. Some IP solvers, such as Ilog Cplex, can solve the problem optimally using the IP model although they are not applicable to medium-scale instances (such as 20 customers), which usually occurs in practice. We therefore do not compare our approach with Cplex.

4.1 Test Data Generation

We create the test instances by adding a load upper bound for each road segment in the 56 classical VRPTW instances generated by Solomon. Of course, the easiest way would be random generation. However, when considering a realistic situation, we can observe that the geographical environment cannot vary entirely in nearby areas. For instance, generally cities are located near to suburbs or rural areas instead of deserts or rain forests. Therefore, we invent the following mechanism to generate the load upper bounds for each instance.

First, the whole map is divided into grids with a small edge length. Each grid is associated with a parameter, $k(0 < k < 1)$. The greater k is, the larger the impact a leak is. For example, the k value is large if the grid is located in an urban area. Therefore, if a road segment passes through n grids G_1,G_2, ..., G_m, the load upper bound is $Min\{q/k_i, 1 \leq i \leq m\}$, where q is a constant. The Min operator is adopted in order to ensure safety along the whole road segment when it passes through multiple grids.

Although the parameter k associated with each grid is randomly generated, if the above-mentioned geographical environment is taken into account, we control the values so that the difference between them in adjacent grids is no greater than a certain value. The detailed algorithm is omitted due to space limitations.

4.2 Results & Analysis

4.2.1 Application to the Hazmat Transportation Problem

To measure the performance of the TS algorithm we proposed, two sets of instances, $H1$ and $H2$, are generated according to the above-mentioned scheme. Each set contains 56 instances that are modified based on Solomon's VRPTW benchmark instances [11]. We keep all the values in Solomon's instances except adding upper bounds on each edge. The average upper bound of $H2$ is about twice the one of $H1$.

To observe the performance of different components of the TS algorithm we propose, four variants of TS are implemented and tested:

- **TS1**: Infeasible solutions are not permitted;
- **TS2**: New path creation and path decomposition operators are not applied;
- **TS3**: Path decomposition operator is not applied;
- **TS4**: All components are included.

The results are given in Tables 1 and 2. The 56 instances in each set are classified into 6 classes, C1,C2,R1,R2,RC1,RC2, which identify different properties of the test cases (Clustered, Random, Clustered-Random). For each variant, we list the average number of vehicles (NV), average distance

(Dis) and average running time (Time, in seconds) for each class of instances. For each set, the minimum results have been marked in bold face.

It is obvious that TS4 is the best algorithm among the variants. It achieves the minimum number of vehicles in all sets of instances compared with other algorithms, with similar running times. It does not achieve the minimum distance because other algorithms use more vehicles. The results obtained by TS1 are vastly different from those obtained by TS4 (about 0.5 vehicles per instance), which shows the importance of allowing infeasible solutions during the search process. The performance of TS2 and TS3 is worse. This comparison clearly demonstrates the necessity of the new path creation and path decomposition operators.

4.2.2 Application to the VRPTW Problem

It is also easy to apply the proposed TS algorithm to the classical VRPTW problem by simply setting each upper bound to be infinity (or the vehicle capacity C in the implementation). We run through the 56 instances and compare the results obtained by our TS4 algorithm with the current best solutions (the best is taken from the VRPTW web site http://w.cba.neu.edu/ msolomon/problems.htm), Sweep, I_1,I_2 (these three are taken from [11]) and the heuristic $NewI_2$ from [12], which was originally used to solve the bi-objective hazmat transportation problem. Table 3 shows the details.

The proposed heuristic is competitive compared with other algorithms. Especially in the C class instances, our algorithm is able to assign the same number of vehicles as the current best solutions. Although the results still have some gaps compared with the current best solutions obtained from various mature algorithms, they are much better than several well-known heuristics. We must remember that our heuristic is not designed for the VRPTW problem. Despite this, our algorithm outperforms the other hazmat transportation algorithm when solving the VRPTW instances. We believe it could provide some insights to improve VRPTW results in the future.

5. CONCLUSION

The safe transportation of hazmat is becoming more important. Although this field has been developing for more than 10 years, only one paper addresses the problem by relating it with the famous vehicle routing problem with time windows. The model developed in that paper is not practical and the solution methods are also faulty. In this paper, we propose a new model for this problem by adding load upper bounds on the road segments in the classical VRPTW model. A sophisticated tabu search algorithm is developed to tackle the problem. Novel neighborhood operators such as dynamic penalty mechanism, penalty regression and other components are devised. Two data sets that are generated from a simulation based on real maps are used to test the performance of our TS and show the functionalities of the different TS components. The algorithm is also applied to the classical VRPTW benchmarks and it outperforms some well-known heuristics and is extremely near to the current best results. This paper proposes a new direction in hazmat transportation research and also contributes some insight to the VRPTW problem.

6. REFERENCES

[1] J. Brandao. A tabu search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, page to appear, 2003.

[2] E. Erkut and A. Ingolfsson. Catstrophe avoidance models for hazardous materials route planning. *Transportation Science*, 34(2):165–179, 2000.

[3] E. Erkut and A. Ingolfsson. Transport risk models for hazardous materials: revisited. *Operations Research Letters*, page to appear, 2004.

[4] E. Erkut and V. Verter. Modeling of transport risk for hazardous materials. *Operations Research*, 46(5):625–642, 1998.

[5] F. Glover and M. Laguna. *Tabu Search.* Kluwer Acadamic Publishers, 1997.

[6] B. Y. Kara and V. Verter. Designing a road nework for hazardous materials transportation. *Transportation Science*, 38(2):188–196, 2004.

[7] J. Karkazis and T. B. Boffey. Optimal location of routes for vehicles transporting hazardous materials. *European Journal of Operational Research*, 86:201–215, 1995.

[8] P. Leonelli, S. Bonvicini, and G. Spadoni. Hazardous materials transportation: a risk-analysis-based routing methodology. *Journal of Hazardous Materials*, 71:283–300, 2000.

[9] G. F. List, P. B. Mirchandani, M. Turnquist, and K. G. Zografos. Modeling and analysis for hazardous materials transportation: Risk analysis, routing/scheduling and facility location. *Transportation Science*, 25(2):100–114, 1991.

[10] J. Luedtke. Hazmat transportation and security: survey and directions for future research. *Technical Report, Department of Industrial and System Engineering, Geogia Institute of Technology*, 2002.

[11] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with the time window constraints. *Operations Research*, 35:254–265, 1987.

[12] K. G. Zografos and K. N. Androutsopoulos. A heuristic algorithm for solving hazardous materials distribution problems. *European Journal of Operational Research*, 152:507–519, 2004.

[13] K. G. Zografos and C. F. Davis. Multi-objective programming approach for routing hazardous materials. *Journal of Transportation Engineering*, 115(6):661–673, 1989.

[14] K. G. Zografos, G. M. Vasilakis, and G. M. Giannouli. A unified framework for developing dds for hazardous materials risk management. *Journal of Hazardous Materials*, 71:503–552, 2000.

Instance	TS1			TS2			TS3			TS4		
Class	NV	Dis	Time	NV	Dis	Time	NV	Dis	Time	NV	Dis	Time
H1-C1	21.11	1853.45	68.22	21.22	1814.16	36.56	21.22	1805.99	49.67	**20.89**	**1799.50**	67.00
H1-C2	21.13	1934.75	67.63	21.13	1872.25	35.38	**21.00**	1855.25	48.88	**21.00**	**1844.94**	67.50
H1-R1	18.67	1645.38	83.08	19.33	**1604.73**	47.33	19.42	1605.84	60.25	**18.25**	1624.04	73.00
H1-R2	17.18	1580.49	67.00	17.55	1562.06	36.82	17.45	**1538.25**	48.09	**16.36**	1596.17	62.64
H1-RC1	19.75	2025.73	81.50	19.75	1995.17	44.13	19.88	1983.59	55.38	**19.50**	**1979.31**	75.00
H1-RC2	19.00	2030.44	69.63	19.00	1947.47	38.50	19.13	**1946.03**	49.75	**18.38**	2022.85	69.13

Table 1: Experimental Results for the Data Set $H1$

Instance	TS1			TS2			TS3			TS4		
Class	NV	Dis	Time	NV	Dis	Time	NV	Dis	Time	NV	Dis	Time
H2-C1	10.33	1018.54	40.22	10.44	929.42	20.78	10.33	925.44	28.44	**10.00**	**906.61**	40.11
H2-C2	5.50	915.38	35.75	6.00	884.05	14.88	6.13	**873.54**	20.00	**5.00**	926.77	34.75
H2-R1	13.42	**1303.40**	48.08	13.92	1310.83	15.17	14.25	1317.77	25.58	**12.75**	1308.33	38.33
H2-R2	5.82	1111.31	36.09	6.82	**1049.37**	14.09	6.82	1078.90	19.82	**5.18**	1154.11	34.36
H2-RC1	13.13	1496.87	48.13	13.50	1479.71	20.13	13.88	1501.43	30.63	**12.75**	**1464.68**	42.00
H2-RC2	5.88	1323.10	36.00	6.13	**1281.43**	13.75	6.50	1293.43	20.13	**5.63**	1318.31	33.50

Table 2: Experimental Results for the Data Set $H2$

Instance	Best		TS4		Sweep		I_1		I_2		$NewI_2$	
Class	NV	Dis	NV	Dis	NV	Dis	NV	Dis	NV	Dis	NV	Dis
C1	10.00	828.38	10.00	930.38	10.0	940.8	10.0	951.9	10.1	1049.8	10.44	1150
C2	3.00	589.86	3.00	649.84	3.0	711.9	3.1	692.7	3.4	921.5	3.09	708
R1	11.92	1209.89	12.75	1267.85	14.6	1449.7	13.6	1436.7	14.6	1638.7	13.25	1367
R2	2.73	951.91	3.00	1060.96	3.2	1448.6	3.3	1402.4	3.3	1470.7	3.09	1261
RC1	11.50	1384.16	12.63	1455.62	14.9	1804.5	13.5	1596.5	14.2	1874.4	13.25	1557
RC2	3.25	1119.35	3.50	1282.98	4.0	1735.7	3.9	1682.1	4.1	1797.6	3.6	1517

Table 3: Experimental Results for the VRPTW Benchmark Instances