A two-phase model based DSS for Grain Dispatching and Transportation

***Haihong Yu*†,††*, Jigui Sun*†,††*, Zehai Li*†,††**

†School of Computer Science and Technology, Jilin University, Changchun, 130012 China
††Key Laboratory of Symbolic Computation and Knowledge Engineer of Ministry of Education, Jilin University, Changchun, 130012 China

Summary
The problem of grain dispatching and transportation is a real vehicle routing problem. This paper presents a two-phase optimization model for solving the problem. A set of the feasible routes is generated in the first phase by means of the graph-search algorithm. The search tree is pruned according to the constraints. An integer programming model is designed to select the optimum routes from the set of feasible routes in the second phase. The total cost of all the optimum routes is the least. Based on the two-phase approach, a decision support system for grain dispatching and transportation was implemented. The system was tested with some real instances. Generally, the routes obtained decreased from 9% to 15% in the distance and reduced from 10% to 18% in operational costs.

Key words:
grain dispatching and transportation, vehicle routing problem, graph-search algorithm, integer programming, decision support system

1. Introduction

Grain is the very important economic and strategic material of a country. The government often needs to delivery grain from a center grain depot to many destination cities. It is necessary for grain managers to lower operational costs and obtain higher efficiency. A decision support system for grain dispatching and transportation can work for grain managers.

The problem of grain dispatching and transportation can be described as follows: There is a set of N destination cities with deterministic demands d_i, $i=1,\ldots,N$, which are served from a central grain depot with a fleet of t delivery trucks of capacity Q. Generally, the objective is to minimize the total distance traveled by the truck fleet, but it is also common to minimize route costs. This problem is a vehicle routing problem(VRP).

In general, there are two ways to deal with the VRP and its variations. One way is the exact methods such as integer programming. Another way is the heuristics approaches or the meta-heuristics approaches such as genetic algorithms or Tabu Search.

Regarding exact approaches, In Ref[1], a mixed integer programming model and a dynamic programming method were proposed, which was applied to a real problem of fuel dispatching in an agribusiness firm where multiple deliveries are considered. Both methods are compared with favorable results for the dynamic approach. However, the problem instances tested range from 5 to 9 destinations. The problem of grain dispatching and transportation illustrated in this paper considers a larger number of 70 destinations. An interesting modification of the classical Branch and Cut algorithm was founded in Ref[2], however the algorithm proposed only offers a solution for a simple variation of the VRP, the symmetric capacitated routing problem. Some synthetic instances was managed to solve to optimality but at a rather high computational cost.

A review of models and exact techniques for the VRP was presented in ref[3], and several branch and bound approaches for the VRP are reviewed and evaluated. Although some of the methods are suitable for fairly sized instances, to the best of our knowledge, none of them deal with problems other than the basic version of the VRP (only vehicle capacity is considered).

As we can see, there are some serious drawbacks about exact methods. Exact algorithms are not suitable for real instances many times since the computational time needed to obtain a solution are unviable and the VRP variations considered are far too simple for the methods to be of any practical use. Furthermore, it is not always easy to adapt integer programming models or Branch and Bound algorithms to more complicated VRP variations.
Heuristic approaches are used when finding an optimal solution for larger instances in a reasonable amount of time is not deemed possible. There are many heuristic and meta-heuristic methods available. A general improvement heuristic, called simulated trading, was applied in a complex parallel computing environment with promising results in Ref[4].

A Tabu Search algorithm to solve the standard VRP was used in Ref[5]. Another meta-heuristic applied to the VRP is given in Ref[6]. In this case an Ant System is implemented but only for a very simplistic VRP with a

Manuscript revised January 2006.

homogeneous vehicle fleet. A Tabu Search algorithm for the VRP was considered, but again, only capacity and route length restrictions are considered in the problem. A kind of over-constrained vehicle routing problem is described in Ref[7], and the problem was transformed to a partial constraint satisfaction problem to be resolved. A Tabu Search algorithm was also used for solving this special kind of problem. In Ref[8],an improved genetic algorithm is proposed based on the novel crossover operator. This algorithm can find the optimal or nearly optimal solution to the VRP effectively. An improved LNS algorithm was put forward based on the shortest route priority strategy in Ref[9], which could solve the VRP with long scheduling horizon effectively.

In Ref[10], three different meta-heuristics for the classic VRP are compared. These are descent search, simulated annealing and Tabu Search. In Ref[11], a total of 10 heuristics was compared, in this case, slightly more sophisticated variations of the VRP are considered.

In fact, the application of the techniques appeared in the literature required a great deal of knowledge and expertise because of the existing gap between the problems considered in the literature and the real problems found in practice. A model and an algorithm management system were proposed to provide support in the modeling of more realistic vehicle routing problems. As we have seen, many of the techniques do not provide the flexibility and responsiveness that a real environment needs. Many of the cited references deal with simplistic versions of the VRP, which do not take into account heterogeneous vehicle fleets or other optimization criteria different from total route length minimization, not to mention specific constraints related to grain specific problems such as the limited number of stops per route, the inability of some trucks to reach some cities or the minimum loadings penalties for the trucks.

The main objective of this paper is to develop a simple method for solving a real problem considered, which is the dispatching of grain to other destination cities . Additionally, this paper introduces a computer system for decision making that closes the gap between the method proposed and the satisfactory.

The remaining of the paper is organized as follows: In Section 2, a first attempt to solve the problem with traditional modeling techniques is shown. Section 3 explains the two-phase approach for solving the problem. In the first phase, graph search algorithm generates every feasible route for the problem. In the second phase, an integer programming model chooses the best set of routes from the feasible route set. In Section 4, a description of the developed DSS is given, along with some results and tests. Finally, in Section 5 some conclusions on the study are provided.

2. The traditional model

Given the previous problem statement, an initial approach used for solving the problem was to use a simple integer programming model. The model used for solving the problem is explained below.

Objective function: $\min \sum_{i=0}^{N} \sum_{j=0, j\neq i}^{N} C_{ij} X_{ij}$

Constraints:

$$\begin{cases} \sum_{i=1, i\neq j}^{N} X_{ij} = 1, \quad j = 1,2\cdots, N & (1) \\ \sum_{j=1, j\neq i}^{N} X_{ij} = 1, \quad i = 1,2\cdots, N & (2) \\ \sum_{i=0}^{N} \sum_{j=0, j\in S}^{N} X_{ij} \leq |S| - 1, & (3) \\ \sum_{i=0}^{N} \sum_{j=0, j\in T}^{N} X_{ij} \leq |T| - k, & (4) \end{cases}$$

Parameters:

Q: capacity of the vehicle;

N: number of destination cities; or truck stops;

d_i: demand of destination city *i*; *i* > 0;

C_{ij}: distance between city *i* and city *j*.

Variables:

$$X_{ij, i\neq j} = \begin{cases} 1, & \text{if a truck goes from city i to city j} \\ 0, & \text{otherwise} \end{cases}$$

where $i, j \in \{1,2,\cdots, N\}$ being 0 the origin or depot.

We can see that the objective function is minimizing the total distance traveled. Constraints (1) and (2) ensure that every city is visited by a truck and that every truck leaves each city. Constraint (3) eliminates sub-tours, i.e. tours that do not start and finish at the grain depot, this constraint is added for every possible subset S of cities, not including the depot. Finally, constraint (4) takes into account the load of the trucks, disallowing overloads; this constraint is added for every set of cities T , including the depot, which constitutes more than a possible truckload (every set that satisfies $\sum_{i\in T} d_i > Q$) and k is the minimum number of cities that have to be taken from T to avoid overloading.

The model does not take into account all the aspects of grain dispatching and transportation problem. We tried this model first, as an initial attempt to evaluate the following possibilities. The model was implemented and solved with LINGO. The model proposed proved to be inappropriate for many reasons, some of which can be resumed as follows:

(1)Inefficiency: The model uses N2 variables and needs an unacceptable amount of computer time in order to find an optimal solution, as a matter of fact, none of the instances tested could be adequately solved.

(2)No Complete: The model does not take into account all the significant aspects of the problem, and this situation affects the measure of effectiveness. In particular, some restrictions of the problem are not satisfied, such as the different truck sizes and their inability to reach some cities.

(3)Objective function: Minimizing the total distance traveled is not exactly the destination, we need to minimize the real cost of the routes. This is one of the major drawbacks of the model since, as we have seen in Section 1, there is a complex route cost function. Minimizing the total distance traveled leads to many short routes with half-empty trucks, which are expensive. Once again, we would have needed a more sophisticated model in order to minimize the real cost of the routes.

3. The two-phase approach

Regarding the problem of grain dispatching and transportation, a big number of stops would lead to a very big number of the possible routes. For example, the capacity of each truck is 30 tons. The minimum demand weight is 4 tons. A truck cannot visit more than six destination cities hence, the number of possible routes for 50 destination cities and six stops per route is:

$$C_{50}^{1}+C_{50}^{2}+C_{50}^{3}+C_{50}^{4}+C_{50}^{5}+C_{50}^{6}=11{,}701{,}202{,}500.$$

That accounts for the possible route variations of six stops, five stops, and so on. Obviously, this is a very big number. However, only a small fraction of the possible routes are actually feasible. For example, a possible route can be one visiting six cities located very far from each other and each one demanding 30 ton of grain. The resulting route would require a truck carrying 180 ton of product and probably traveling for more than 700km. This is clearly an infeasible route.

A close examination of the real data leads to the following conclusions:(1) The number of routes with six destination cities is very low, not every route can have six stops, since the maximum allowed load of the trucks is limited; and, as we have seen, we can only serve one cities from one compartment and the number of compartments is fixed in each truck.(2) A route can not be longer than 3000 km which also limits the number of routes with many stops.(3) Some trucks can not visit some cities due to constraints in weight, small roads, bridges or tunnels.

Taking into account these constraints, we found that for the previous example, the actual number of feasible routes was only 1826.

Therefore, we consider that graph search algorithm is used to obtain all the feasible routes. Then a new model of integer programming is built up to solve the problem. Since the new model has much less variables, it must be in the reasonable computational time to get the optimal solutions.

Let us consider only three destination cities, named C1, C2 and C3. The grain requirements are 30, 18 and 9 ton respectively. The distance between these three cities and the grain depot can be seen in Fig 1.

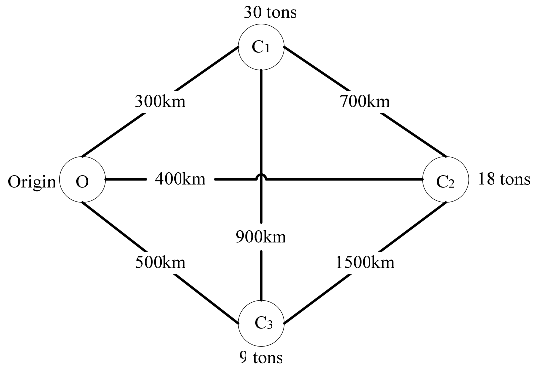


Fig.1. An example of the grain dispatching and transportation with 3 destination cities

The graph search algorithm starts from the origin v_0. Then it goes to the next node v_1, which is linked to v_0 by an edge. The algorithm evaluates the route $v_0 \to v_1 \to v_0$. The route $v_0 \to v_1 \to v_0$ is saved if the route is feasible. Then it goes to the next node v_2, which is linked to v_1 by an edge. The algorithm evaluates the route $v_0 \to v_1 \to v_2 \to v_0$, and it goes on. Once it finds that the route $v_0 \to v_1 \to \cdots \to v_i \to v_0$ is unfeasible, it stops. The pruning happens. It goes back to node v_{i-1}. Another node different from v_i is chosen, and it goes on. If there is no different node to choose, it goes back to the previous node of v_{i-1}. And the procedure aforementioned is repeated again. It comes back to v_0 finally. The algorithm ends if all the nodes linked to v_0 are visited. All feasible routes are saved. The distance of each route and the load of each truck are computed.

let us consider this example; We take city C_1, then one feasible route is $O \to C_1 \to O$, traveling for 600km and carrying 30 tons of grain, then we take that generated route and see if we can generate more feasible routes by adding more stops. From city C_1, we cannot go to city C_2, for the resulting route $O \to C_1 \to C_2 \to O$ would carry 48 tons of grain and this would not be feasible. So, the $O \to C_1 \to C_2 \to O$ route is rejected and therefore, we do not add more stops to this route since it is not generated. This methodology ensures that the tree depicted in Fig.2 is adequately pruned.

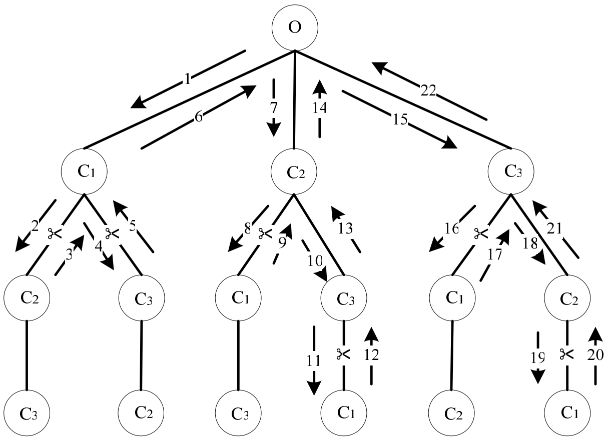


Fig.2. The running procedure of graph search algorithm for the example

For this example, the search tree that the algorithm generated is in Fig.3. Table 1 shows all the feasible routes in this example.

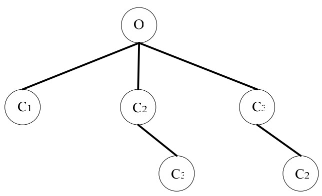


Fig.3. The search tree the graph search algorithm generated for the example

Table 1 . All the feasible routes in this example

No.	1st stop	2nd stop	3rd stop	Route	Distance (km)	Loading (ton)
1	C1			O → C1 → O	600	30
6	C2			O → C2 → O	800	18
9	C2	C3		O → C2 → C3 → O	2400	27
11	C3			O → C3 → O	1000	9
14	C3	C2		O → C3 → C2 → O	2400	27

The graph search algorithm gives, for each feasible route, the set of destination cities it visits, the total distance traveled, the total cost of the route, and the kind of truck to be used. Once all feasible routes are generated, we need a model able to choose the best set of routes from the feasible route set; within this best set, every destination cities has to be visited exactly once. The objective is minimizing the total cost of the routes, which is now known. This model is explained below.

Set $L=\{l_1,l_2,\cdots,l_n\}$ is the set of feasible routes. n is the number of feasible routes. Set $C=\{c_1,c_2,\cdots,c_m\}$ is the set of destination cities. m is the number of destination cities.

Set

$$a_{ij}=\begin{cases}1, & \text{if the route } l_i \text{ visits city } c_j\\ 0 & \text{otherwise}\end{cases},$$

$Cost_i$=cost of the route l_i,

$$x_i=\begin{cases}1, & \text{if the route } l_i \text{ is done}\\ 0, & \text{otherwise}\end{cases}$$

Where i 、 j is natural number, and $1\le i\le n$, $1\le j\le m$

Objective function: $Min\sum_{i=1}^{n} Cost_i x_i$

Constraints: $\sum_{i=1}^{n} a_{ij}x_i=1$, j is natural number , and $1\le j\le m$

As we can see, there are much less variables in the model. All parameters are calculated by the graph search algorithm, so any changes in specification could be done in the algorithm without needing to change the model. This model proved to be extremely fast and robust, the solution times were always under 60s for normal tests, and under 300s for models with 20,000 to 30,000 binary variables. We found that the use of the different constraint cuts reduced the computing time;

In Table 2, there is a comparison between the traditional model and the two-phase model. We see that, for the previous five simplified instances, the two-phase model provides solutions that are always less expensive, even when the routes obtained travel a slightly longer distance.

Table 2 . The comparison between the traditional model and the two-phase model.

No.	traditional model			Two-phase model		
	Distance	Cost	Running	Distance	Cost	Running

	(km)	(￥)	time(s)	(km)	(￥)	time(s)
1	2178	3664	44.5	2251	3519	5.0
2	2217	3589	30.8	2255	3481	4.0
3	1554	2680	70.9	1532	2578	12.0
4	2832	4163	281.7	2912	4032	78.0
5	2447	3982	117.1	3421	3849	16.0

4. The Decision Support System

Once the two-phase model proved to work, we needed to develop a software application to facilitate grain managers to use it. We designed and implemented a DSS for grain dispatching and transportation (GDT-DSS) with Microsoft visual C++6.0 and Microsoft SQL Server2000.

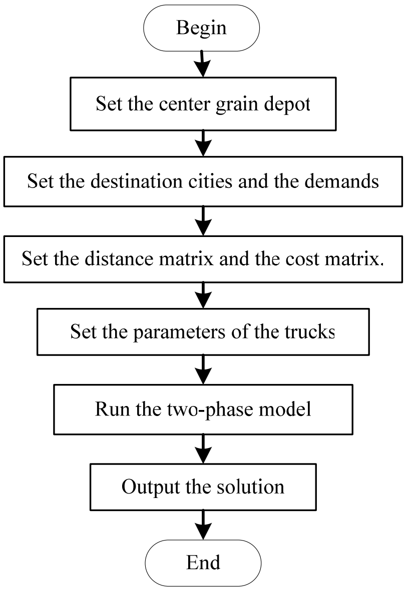


Fig.4. The operating flow of GDT-DSS

In order to develop a flexible DSS, we needed to consider every aspect of the route scheduling process. Fig.4 shows the operating flow of GDT-DSS. The solution process with GDT-DSS can be explained as follows:

First, the user can set the center grain depot and the set of destination cities. Then the user can input the demands of all destination cities. At any moment, the user can add or modify the demands; additionally, the past demands can be selected, which gives an enormous flexibility to the grain manager. The user can change at any moment the default distance between each city pair or the possibility of traveling between each pair of cities. By doing this, we can consider many situations such as: asymmetric traveling distances, traveling to a city but not back, bad road conditions, bridges, tunnels and more.

After inputting the demands and changing the distance or accessibility data the user can run the solver, a very useful window shows the progress of the different phases of the solution process (Fig.5).

Fig.5. Interface of Solution in process, which shows the current state of the GDT-DSS while solving the problem.

When the model is solved, a solution screen is shown. At this phase, the user can select which routes are good and which are not, he might even add or modify demands. At every phase, data can be changed. The DSS responds to data changes, as well as to condition changes with a color coded button at the toolbar showing that the last optimum solution obtained might have changed under the new conditions. With this tool, the user can make what-if analysis.

We tested GDT-DSS for 10 different instances, which were picked randomly from those of which we had historical data, so we could compare the actual route schedules obtained by GDT-DSS and the routes that were actually scheduled. The results show that the grain management department could have saved from 10% to 18% in operational costs and from 9% to 15% in the distance traveled by trucks. This is a worst-case scenario, because the grain manager could have used GDT-DSS to enhance and refine the solution much better than us, as we only did single run tests, without using all the possibilities of the DSS.

5. Conclusion

In the present paper, we aimed at solving the problem of grain dispatching and transportation by providing a DSS for effective decision making.

The model used by GDT-DSS is simple, in the way that the model is easily understood by the decision maker. The model is also complete, easy to handle, and easy to communicate with other software as well as user friendly. We showed that, given the correct generation of the variables of a model, the problem can be solved, with only minor simplifications, to an optimal solution, in computing times under 1 minute in most of the cases.

The grain manager can now use GDT-DSS and have accuracy and speed over the route scheduling process, allowing him or her to improve the solution by adding them and by re-running GDT-DSS, subsequently obtaining a new optimum solution in a very short amount of time.

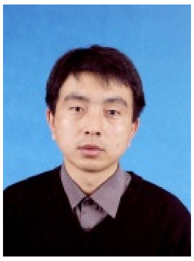
The proposed two-phase approach could be used in other similar transportation problems, Furthermore, the addition of time windows to the problem is trivial, since we would simply need to have a travel times matrix–similar to the distance matrix–and to modify the graph search algorithm in order to take into account traveling times for satisfying time windows.

Acknowledgment

This paper was supported by National Natural Science Foundation of China (Grant No.60273080, 60473003), the Science and Technology Development Program of Jilin Province of China (Grant No.20040526) and the Outstanding Youth Foundation of Jilin Province of china (Grant No.20030107).

References

[1] Onal.H, Jaramillo.B.M., Mazzocco.M.A. Two formulations of the vehicle routing problem:An empirical application and computational experience[J]. Logistics & Transportation Review.1996,32 (2), pp.177–190.
[2] Blasum.U.,Hochstattler.W.Application of the Branch and Cut Method to the Vehicle Routing Problem[J]. Technical report.Zu-Koeln University.2000.http://www.zaik.uni-koeln.de/~paper/index.html?show /zaik2000-386
[3] Toth.P.,Vigo.D.Models,Relaxations and Exact Approaches for the Capacita -ted Vehicle Routing Problem[J]. Discrete Applied Mathematics.2002.123(1-3), pp.487-512
[4] Bachem.A,Hochstattler.W,Malich.M. Simulated Trading: A New Parallel Approach for Solving Vehicle Routing problems[J].Discrete Applied Mathematics.1996.65 (139), pp.47−72.
[5] Xu.J,Kelly.J.P,A network flow-based tabu search heuristic for the vehicle routing problem[J].Transportation Science.1996.30 (4), pp.379−393.
[6] Bullnheimer.B,Hartl.R.K,Strauss.C.Applying the Ant System to the Vehicle Routing Problem[C]. In 2nd International Conference on Meta-heuristics－MIC97. 1997.
[7] LI Ju-fang, TAN Yue-jin.A Partial Constraint Satisfaction Vehicle Routing Problem and It s Solving Algorithm[J].Operations Research and Management Science.2004. 13(5),pp.12-17
[8] ZHANG Li-ping,CHAI Yue-ting.Improved Genetic Algorithm for Vehicle Routing Problem[J]. Theory and Practice of System Engineering.2002.22(8). pp.79-84
[9] Liu Xiao-lan,HAO Zhi-feng,WANG Guo-qiang,FU Ke-qiang.Improved large neighborhood search algorithm for vehicle routing problem with time windows[J]. Computer Integrated Manufacturing Systems. 2004.10(7), pp.825-831
[10] Van Breedam, A.Comparing decent heuristics and meta-heuristics for the vehicle routing problem[J]. Computers & Operations Research.2001.28 (4), pp.289−315.
[11] Van Breedam, A. A parametric analysis of heuristics for the vehicle routing problem with side-constraints[J]. European Journal of Operational Research.2002.137 (2), pp.348−370.



Haihong Yu received the B.E. degree from NanJing University of Science & Technology in 1998, and received the M.S. degree in Computer Science from Jilin University in 2001, and now is a Dr. student in the school of Computer Science and Technology, Jilin University. His research interest is decision support system, human-machine interface, natural language processing.

Jigui Sun received M.S. degree in Mathematics from Jilin University in 1988 and received the Dr. degree in Computer Science from Jilin University of China in 1993, and was promoted professor in July 199**7.** He is currently head of Ministry of Education Key Laboratory of Symbolic Computation and Knowledge Engineering and associate dean of College of Computer Science and Technology, Jilin University. He is director of China Computer Association and committee member of the sub-division of computer science and technology for guiding higher education of the Ministry of Education. He is the Ministry of Education's Distinguished Talent for the New Century. His research work focuses on artificial intelligence, constraint programming and intelligent information processing.

Zehai Li received M.E. degree in Electronics from Jilin University in 2001, and now is a Dr. student in the College of Computer Science and Techno- logy, Jilin University. His research interests include data warehouse, OLAP and decision support system.