Tabu Search and Greedy Algorithm Adaptation to Logistic Task

Kamil Musial, Joanna Kotowska, Dagmara Górnicka, Anna Burduk

Tabu search and greedy algorithm adaptation to logistic task

Kamil Musiał, Joanna Kotowska, Dagmara Górnicka, Anna Burduk

Wrocław University of Science and Technology, Faculty of Mechanical Engineering, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{kamil.musial, joanna.kotowska, dagmara.gornicka, anna.burduk}@pwr.edu.pl

Abstract. Distribution companies, in order to maintain a competitive advantage, must demonstrate not only the quality of the offered goods, but also the speed of execution of orders. This article deals with the allocation of the available capacity of transport during the transportation of goods between companies. Solving the problem of optimization is offered by the chosen methods of an artificial intelligence, such as Tabu Search algorithm, greedy algorithm and Tabu Search using the results of the greedy algorithm. The results were compared with the actual results of one of the Dutch distribution companies.

Keywords: Tabu Search, Greedy, Optimization, VRP

1 Introduction

Nowadays distribution companies are required to manage such transport processes that the orders are delivered on time at the lowest possible cost. However, most of the decisions about the optimization problem is taken only on the basis of human beliefs. Such an approach, however, often result in skipping the best solution, while generating financial losses. A similar problem was observed in one of the Dutch distribution companies. This paper addresses the issue of possible use of available transport capacity during the transportation of products. Finding the most favorable solution to this type of problem is possible if using methods of linear programming. However, the specificity of distribution companies as well as a variety and often a large number of orders forces reduction of the time required for the calculation. Artificial intelligence methods gained more and more popularity in recent years in the search of solutions close to optimal in a relatively short period of time in different areas.

The vehicle routing problem (VRP) is undoubtedly one of the most studied combinatorial optimization problems in literature. It can be defined as the problem of designing routes for delivering vehicles of given capacities, to supply a set of customers with known locations and requests from a single depot [11, 12]. Routes for the vehicles are designed to minimize some objective such as the total distance traveled [7]. The VRP is NP-hard and therefore most of the literature has focused on heuristic

adfa, p. 1, 2011.

solution methods [3, 4, 5, 8]. Vehicle routing problems exhibit an impressive record of successful meta-heuristic implementations [9].

Meta-heuristic algorithms can be applied on many kinds of decision problems without the need to transform them into mathematical formulations [1, 2, 9]. Tabu search is a meta-heuristic approach that guides a local heuristic search procedure to explore the solution space beyond local optimality [4, 5, 6]. One of the important characteristics is memory function, which can restrict some search directions for a more detailed local search and make it easier to leave local optimum solution. In tabu search, a tabu list is used to define search prohibition, and a move is a process that the algorithm uses to change the current solution to the new solution. In this research, the greedy randomization adaptive search procedures are adopted to generate alternative moves, and these dynamic moves are designed to control the search direction [10]. The idea of dynamic move range tends to reduce search space near a good move; otherwise, the search space is enlarged. The greedy design means that the best candidate move is compared with all tabu moves, and good moves are removed from the tabu list. The good moves refer to the moves that are better than the current best solution. During the simulation, vehicles' attributes and paths are assumed fixed [7].

2 Problem description and notation

The analyzed case concerns the Dutch company that distributes clothing stores in Denmark, Belgium and France. Various types of clothing are transported in cartons of a particular volume or on hangers with a volume depending on the type of clothing. 16 couriers is responsible for the carriage of goods. They have private cars with a specified maximum transport capacity. Each carrier supports the route assigned to him, without possibility to change it. Commodity turnover occurs rapidly, and transportation plans are adjusted up to date, depending on orders. Due to the pace of the generated orders there are situations where it is impossible to implement all of them, because it exceeds the available capacity of a transport. Then, the order of their execution determines the priority of the order (Table 1).

Table 1. Orders priority

Colors	Name	Case
Green	Normal	Date of implementing the order is relatively distant
Yellow	Increased	Order with a fixed time is required due to the fact it has been exceeded before for this particular client
		Delivering the order on time is crucial for the bigger purchase order to end successfully
		Implementation Deadline passes within 1 day.
		Thera are expected difficulties in implementing of the order
Red	High	The contract is executed for a client with a personalized strategic agreement.
		Exceeding the delivery time is associated with high financial penalty

		Order is executed again due to the error caused by the company.
		The final term of the execution of the order passes that day.

The „red” orders are implemented as first with the highest priority, "yellow" orders should be given higher priority than the "green" but they are not critical for the day.

The company is growing and thus the appropriate management of order is more challengeable. Solutions used before become insufficient when customers required speed when at the same time a number of packages and their sizes were rising. The desire to preserve the competitive advantage of the company requires optimizing the ways of transporting goods to contractors.

3 Characteristics of optimization model

3.1 Issue analysis

Computer simulation of random solutions allows finding the optimum solution for these types of problems. The program introduced the actual data of relevant cases for the ability to compare the results obtained with the solutions chosen by the company. The model was built on the given example of the existing distribution company. The data come from the actual procurement plan for the day. Table 2 shows a selected fragment of the couriers daily work summary.

Table 2. Daily summary excerpt

No.	Capacity [m^3]	Amount of available orders [pcs]	Orders handled [pcs]	Used capacity [m^3]	% of max capacity	Yellow orders [pcs]	Red orders [pcs]
1.	8	14	4	7.464	93.3	1 of 2	N/A
2.	12	22	8	10.72	89.33333	3 of 4	2 of 2
3.	13.5	11	8	12.94	95.85185	2 of 2	N/A
4.	9.2	23	5	9.161	99.57609	1 of 1	1 of 1
5.	18	17	12	17.664	98.13333	3 of 3	N/A
6.	22	12	12	13.413	60.96818	1 of 1	N/A
7.	14.4	15	14	14.37	99.79167	5 of 5	3 of 4
8.	11	8	8	9.926	90.23636	2 of 2	N/A
9.	9	14	6	8.527	94.74444	1 of 1	2 of 2
10.	14.3	15	10	14.066	98.36364	1 of 1	N/A
11.	14	6	6	3.826	27.32857	2 of 2	N/A
12.	7	11	7	6.841	97.72857	1 of 1	N/A
13.	12	16	8	10.384	86.53333	6 of 8	1 of 1
14.	7.2	11	5	6.955	96.59722	4 of 4	N/A
15.	16	21	10	14.482	90.5125	4 of 6	1 of 2
16.	11.4	12	6	10.58	92.80702	1 of 3	N/A
		228			94.86715	-8	-2

16 courier has from 6 to 22 contracts, in total – 228 contracts with an average volume of 1.145m3, including 47 of the yellow priority and 12 with red priority. Capacities of cars is from 7 m3 to 16m3. On that day 3 couriers had the volume of orders less than the maximum capacity of their cars. In this case, each of the test algorithms shows the optimal solution, and the percentage use of the available capacity of the car is not related to the type of optimization, and therefore these examples are not taken into account in further analysis.

3.2 Model assumptions

The main function of the program was to establish the maximum utilization of the available capacity in the way the supply of goods is the most favorable. Constant limits assumed in the case are:

- the volume of garments in cartons,
- the volume of clothes on hangers,
- the maximum capacity of the transport courier car,
- the number of couriers.

Size volume introduced to the nearest 1 dm3 (1L). In addition, the following assertions were made:

- routes are assigned to specific cars and couriers, there is no possibility of converting cars or route changes,
- mass of the goods is irrelevant, because the clothing is relatively lightweight and is not a critical factor in the present case,
- carriers cannot share orders between them,
- ruled out the possibility of making individual arrangements with couriers.

(Eg. returning for the remained orders in order to save the length of the route, at the expense of realization of the remaining orders later).

3.3 Introduced optimization parameters

For the purpose of the algorithm was introduced parameter "p", stating the importance of the group of contracts, adequately to the priority. After analyzing the results of a delivery the priorities have been assumed. The orders for the "red" parameter "p" accepts the importance of 1000, for the procurement of "yellow" 3 and for "green" contracts 1. Priority red are absolutely a priority, and giving priority to orders of yellow scales with a value of 3 enables to treat them in the first place, if it does not drastically reduce the use of space.

4 Considered algorithms

For the considered case it was decided to use algorithms: Tabu Search, Greedy algorithm and a combination algorithm Tabu Search using the solution proposed by the greedy algorithm as a zero solution / start solution. The aim was to present the company calculation method allowing to find a solution in 5 seconds or less using a mid-range computer. This will allowing the company to freely modify orders and receive converted solutions in near real-time. The programming language used to create the algorithms is C++.

4.1 Tabu Search

Tabu Search is an algorithm that searches the space created from all possible solutions with a particular sequence of movements. Among them, there are taboo movements (forbidden). The algorithm avoids oscillation around the local optimum by storing information about proven solutions in the form of a list of taboos (TL). Tabu Search algorithm is a deterministic, in a contrast to e.g. the genetic algorithm. Treating the same data several times with the same Tabu Search algorithm will give the same results.

```
Algorithm Tabu Search (S0, var S, max_m, max_iter)

Set S = S0 and iter = 0
Repeat
        m = 0
        best = 0
        iter = iter + 1
                Repeat
                        m = m+1
                        Execute Check_the_neighboring_solution (S,Sm)
                        Execute Check_Tabu_list (S,Sn)
                                if (f(Sm) > best then (best = f(Sm) and (m2= m))
                until (m = max_m)
        Execute Add_to_Tabu_list (S,Sm2)
        S= Sm2
        If best > sol then sol = best
Until iter = max_iter
```

Analyzed problem can be reduced to a binary sequence:

1 1 0 0 1 0 1 ...

in which each bit corresponds to the order assigned to the courier. The value of "1" means that the courier undertakes the contract, the value of "0" that he rejects it. For example, the implementation of the five orders starts from the obtaining of the set of orders, described as a specified bit sequence, eg:

1 0 1 0 1.

<u>Check the neighboring solution</u>:
The algorithm checks all neighboring solutions (differing in one bit):

1 0 1 0 0

1 0 1 1 1

1 0 0 0 1

1 1 1 0 1

0 1 1 0 1 ,

<u>Check Tabu List:</u>

The algorithm checks whether the transition between data solutions has already been made. If so, such a solution is skipped and then the best solution, according to the specified criterion, is chosen:

1 1 1 0 1

<u>Add to Tabu List:</u>

And adds it to a list of prohibited movements (TL): If a found solution is the best found so far, the algorithm remembers it as the solution algorithm. The algorithm then repeats the operation with specified number of times

In the case of each bit represents the size of the contract [o]- order [o1, o2…..on] and its priority [p1, p2, p3]

According to the assumptions (Section 3.2) capacity of transport vehicles for the route is strictly defined. The algorithm has a limit: the total volume of contracts multiplied by the bit value representing the order does not exceed the maximum capacity of the transport car Vmax.

$$0o_1 + 1o_2 + 1o_3 + \cdots + 0o_n \leq V_{max} \qquad (1)$$

Following this equation it is seen that the total volume of the selected orders cannot exceed the maximum available capacity of the car.

According to the adopted objective function (Section 3.2) criterion for choosing the best solution is the total volume of contracts multiplied by the priority values and by the priorities of the value of bits representing the order data.

$$F(c) = 0o_1p_1 + 1o_2p_2 + 1o_3p_3 + \cdots + 0o_np_n \rightarrow MAX \qquad (2)$$

The algorithm was set to 100 iterations.

4.2 Greedy algorithm

Greedy algorithm is a method of determining a solution by making a selection in each step. This means that the chosen solution is the one that seems to be the best at the moment. The rating is locally optimal but does not analyze if the data in the following steps will make sense. The greedy algorithm, like Tabu Search algorithm also does not guarantee finding the optimal solution.

Greedy method (o, p, V, V_max, o_number)

V = 0

Execute **Sort descending**(o,p)

Execute: **Add highest orders** (o,p,V_max)

Repeat:

    Execute: **Add next** (o,p,Vmax)

Untill (V = V_max) **or** (all orders added)

Sort descending:

In this issue algorithm multiplies the volume of each order by the value of its priority, and the sorts the results descending:

$$o_{1'}p_{1'} \geq o_{2'}p_{2'} \geq o_{3'}p_{3'} \geq \cdots \geq o_{n'}p_{n'} \tag{3}$$

Add highest orders:

Thereafter adds capacity orders from the highest sorted values until it reaches the maximum value, not exceeding the capacity of the car:

$$o_{1'} + o_{2'} + \ldots + o_{m'} \leq V_{max} \tag{4}$$

Add next:

When adding another order exceeds the permitted volume / overflow the car, then this solution is skipped, the next is added and the criterion of maximum capacity is checked. If:

$$o_{1'} + o_{2'} + \ldots + o_{m'} + o_{m+1'} > V_{max} \tag{5}$$

Then:

$$o_{1'} + o_{2'} + \ldots + o_{m'} + o_{m+2'} \leq V_{max} \tag{6}$$

The algorithm terminates when the available capacity will be used:

$$o_{1'} + o_{2'} + \ldots + o_{m'} = V_{max} \quad (7)$$

or when all the orders will be added or rejected.

$$o_{1'} + o_{2'} + \ldots + o_{m'} + o_{m+2'} + o_{m+5'} + o_{n-1'} + o_{n'} \leq V_{max} \quad (8)$$

4.3 Solution combining greedy algorithm and Tabu Search.

For the purposes of research a solution combining the above two methods is proposed. The third algorithm starts its operation in accordance with the specifics of the greedy algorithm. The solution obtained in this way is a set of an input data to the search algorithm according to the algorithm method Tabu Search. Functional diagram is shown in Fig. 1.

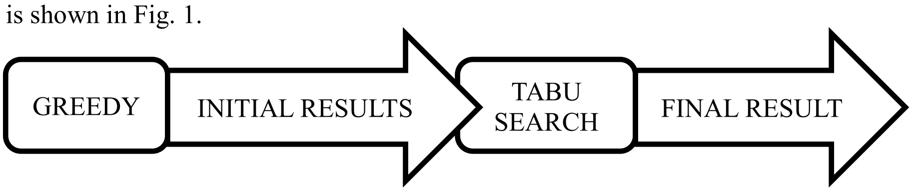

Fig. 1. Diagram of the combined algorithm

Combined algorithm presented in Fig. 1. has been analyzed regarding to the accuracy of this solution according to the accepted criterion in comparison with a simple algorithm Tabu Search and the number of iterations needed which affects the time needed for the program to turn the best result.

5 Analysis of the results

Based on the assumptions pictured algorithms were developed for the analyzed case was. The use of analyzed algorithms to solve the optimization of a problem has been compared. Their effectiveness has been studied on the example of a real case of status orders for the one day in the analyzed company. Each of the algorithms proposed the most beneficial plan for the transport of distributed goods. The hardware used to make the computation is a medium-class laptop with processor Intel Core i5-430m, 8GB RAM and Windows 7 Professional, 64-bit Operating System. The results obtained, in terms of amount of completed priority orders and the use of available capacity is shown in Table 2 below.

Table 2. Summary of results of analyzed methods

Method	% of used capacity	Orders uncompleted yellow	Orders uncompleted red

Current	94,87	8 / 47	2 / 12
Tabu Search	99,29	0 / 47	0 / 12
Greedy	98,67	1 / 47	0 / 12
Tabu Search + Greedy	99,39	0 / 47	0 / 12

Be analyzing the capacity of used cars it can be observed that each of the proposed method achieves better results than the current system of the company. The difference between the real solution, and the solutions suggested by the artificial intelligence methods in the use of available capacity is less than 5 percentage points, giving a real gain for the company. Couriers, selecting orders in accordance with their experience left 8 orders of yellow priority and 2 of red priority, which is unacceptable in this case. The solution proposed by the greedy algorithm leaves one order of yellow priority – this situation is acceptable. Algorithms, Tabu Search and Tabu Search enriched with solution of greedy algorithm allowed 100% accomplishment of the priority order.

Table 3. Summary of the best iteration

	Number of the best solutions	
No.	**Tabu search with Greedy**	**simple Tabu Search**
1.	0	5
2.	4	15
3.	3	7
4.	0	8
5.	1	17
6.	0	12
7.	3	5
8.	2	12
9.	1	10
10.	2	14
11.	4	15
12.	4	16
13.	0	7

As mentioned in section 4.1 Tabu Search algorithm was set to 100 iterations – it analyzed 100 neighboring solutions and passed to the best.one. As can be seen in Table 3. simple Tabu Search algorithm achieved the best solution after maximum of 17 iterations. This means that if the number of iterations was reduced to 17 and accelerated search solutions around the 5-fold, in a given case, the results would be identical. Tabu Search algorithm starting from the solutions generated by the greedy algorithm needed at maximum 4 iterations to get the best solution. The number of the best iteration of "0" means that the input set of orders was not improved during 100 iteration - Tabu Search algorithm is not able to improve the solution proposed by the greedy algorithm. As shown in Table. 2 modified Tabu Search algorithm showed the best use of the capacity of the car. The algorithm combining the greedy algorithm and Tabu Search is the best in terms of mentioned criteria and provides a good solution for the analyzed case and can give the company a real gain.

6 Summary

For the purpose of research, an algorithm which, by the use of available resources, allows to find the most beneficial way to transport distributed goods has been built. The results of the research conducted on the example of a real Dutch distribution company demonstrated the potential benefits that could be achieved by the use of the proposed methods. Each of them allows to achieve much better results than the current system of the company. The differences between the results achieved by the three tested algorithms are small. For the analyzed case, the most advantageous method proved algorithm combining operation methods of Greedy algorithm and Tabu Search.

7 References

1. Burduk A., Musiał K.: Genetic algorithm adoption to transport task optimization, In: International Joint Conference SOCO'16, Springer, Advances in Intelligent Systems and Computing, vol. 527, pp. 366-375, (2016)
2. Burduk A., Musiał K.: Optimization of chosen transport task by using generic algorithms In: Computer Information Systems and Industrial Management, CISIM 2016, Springer, Lecture Notes in Computer Science, vol. 9842, pp. 197-205, (2016)
3. Baldacci R., Christofides N., Mingozzi A.: An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, Math. Prog. Ser. A, Vol. 115, No. 2 pp. 351–385, (2008)
4. Bożejko W., Uchroński M., Wodecki M., Parallel Tabu Search Algorithm with Uncertain Data for the Flexible Job Shop Problem, Lecture Notes in Artificial Intelligence No. 9693, Springer (2016)
5. Glover, F., and Laguna, M.: Tabu search, Kluwer Academic Publishers, Norwell, MA, (1997)
6. Grabowski J., Pempera J.: New block properties for the permutation flow shop problem with application in tabu search, Journal of the Operational Research Society, Vol. 52, nr 2, pp. 210-220 (2001)
7. Hu T., Chen L.: Traffic Signal Optimization with Greedy Randomized Tabu Search Algorithm, Journal of Transportation Engineering, Vol. 1040, No. 8, (2012)
8. Lysgaard J., Letchford N.A., Eglese W.R.: A new branch-and-cut algorithm for thecapacitated vehicle routing problem, Math. Program, Vol. 100, No.2, pp. 423–445, (2004)
9. Pacheco J., García I., ÁlvarezA.: Enhancing variable neighborhood search by adding memory: Application to a real logistic problem, Knowledge-Based Systems, No. 62 pp. 28–37, (2014)
10. Resende M., Ribeiro C.: Greedy randomized adaptive search procedures, International Series in Operations Research & Management Science, No. 57, Springer, New York, pp. 219–249, (2003)
11. Rudawska A., Čuboňova N., Pomarańska K., Stanečková D, Gola A.: Technical and Organizational Improvements of Packaging Production Processes, Advances in Science and Technology. Research Journal, Vol. 10, No. 30, pp. 182-192, (2016)
12. Toth P., Vigo D.: Branch-and-bound algorithms for the capacitated VRP, in: The Vehicle Routing Problem, SIAM, Philadelphia, pp. 29–52, (2001)