Solving the VRPTW through a Robust Metaheuristic

Thomas Martinod[1]

[1] *Heuristics Course, EAFIT University, Medellín, Colombia*

November 22, 2024

Introduction

Objective: Solve the Vehicle Routing Problem with Time Windows (VRPTW) by combining multiple metaheuristics.

We propose an algorithm that uses the four following metaheuristics.

- **Simulated Annealing:** Provides a strong initial solution through probabilistic exploration and acceptation.
- **Tabu Search:** Refines the solution using memory-based local search.
- **ALNS:** Explores diverse neighborhoods by adaptive destruction and repair.
- **VND:** Given the previous three, finds the local optima based on 6 neighborhoods.

Based on the results of the first homework, we use initial solutions from a constructive method, Reactive GRASP (or simply GRASP) and ACO.

Simulated Annealing (SA)

Simulated Annealing (SA) is an algorithm inspired by the physical annealing process, where a material is heated and then cooled in a controlled manner. In optimization, this translates to:

- The objective function $f(x)$ evaluates the quality of a solution x.
- A **starting temperature** T_0 and a **cooling rate** $\alpha \in (0, 1)$ are defined.

The probability of accepting a worse solution S' from S is given by:

$$P(S \to S') = \begin{cases} 1, & \text{if } f(S') < f(S) \\ \exp\left(-\frac{f(S') - f(S)}{T}\right), & \text{if } f(S') \geq f(S) \end{cases}$$

where T decreases in each iteration according to $T \leftarrow \alpha T$.

In the VRPTW approach, SA is used for an initial broad exploration of the solution space [1].

Tabu Search (TS)

Tabu Search (TS) is an iterative local search algorithm that avoids revisiting solutions by using a memory structure called the **tabu list** [2].

Key Concepts:

- **Tabu List ($\mathcal{T}$)**: Temporarily stores recent moves to prevent cycling.
- **Aspiration Function (A)**: Allows tabu moves if they lead to a significant improvement.

Neighborhood Search: From a solution S, the next solution is selected as:

$$S_{t+1} = \underset{S' \in \mathcal{N}(S) \setminus \mathcal{T}}{\arg\min} f(S')$$

TS dynamically adjusts the tabu tenure (τ) to balance exploration and intensification.

Tabu Search Parameters

Main Parameters:

- **Tabu Tenure (τ)**: Determines the size of the tabu list, controlling how long moves remain forbidden.

$$\mathcal{T} = \{M_1, M_2, \ldots, M_\tau\}$$

- **Objective Function Weights (α, β)**: Balances the trade-off between minimizing total distance and reducing the number of routes:

$$\text{Cost} = \alpha D + \beta K$$

ALNS - Adaptive Large Neighborhood Search

The **ALNS** algorithm is an adaptive strategy that employs **destruction and repair operators** [3] to explore the solution space globally. It selects these operators probabilistically and adjusts their usage frequency based on their performance.

Key components:

$$\mathcal{D} = \{d_1, d_2, \ldots, d_k\} \quad \text{Destruction Operators}$$

$$\mathcal{R} = \{r_1, r_2, \ldots, r_m\} \quad \text{Repair Operators}$$

Iterative process: For each iteration, ALNS selects:

$$d_i \in \mathcal{D} \quad \text{and} \quad r_j \in \mathcal{R}$$

with probabilities proportional to their adaptive scores $p(d_i)$ and $p(r_j)$.

Destruction Operators in ALNS

- **Random Removal** (d_{random}): Randomly removes customers without considering their cost.
- **Worst Removal** (d_{worst}): Removes customers with the highest incremental cost. For a customer i in its route:

$$\text{cost}(i) = c(p, i) + c(i, s) - c(p, s)$$

  where p and s are the preceding and succeeding nodes of i.
- **Route Removal** ($d_{\text{route removal}}$): Deletes entire routes to encourage reinsertion from scratch.
- **Least Utilized Removal** ($d_{\text{least utilized}}$): Removes routes with the lowest total load, freeing underutilized customers for reassignment.

Repair Operators in ALNS

- **Greedy Repair** (r_{greedy}): Inserts each customer at the position with the smallest cost increment:

$$\Delta c = c(i,p) + c(i,s) - c(p,s)$$

where p and s are adjacent nodes in the selected route.
- **Regret Repair** (r_{regret}): Calculates the regret value for inserting a customer by comparing the cost of the two best positions:

$$\text{regret}(i) = \Delta c_2(i) - \Delta c_1(i)$$

where $\Delta c_1(i)$ and $\Delta c_2(i)$ are the costs for the best and second-best positions, respectively.
- **Savings Repair** (r_{savings}): Inserts pairs of customers that maximize cost savings, calculated as:

$$s(i,j) = c(d,i) + c(d,j) - c(i,j)$$

where d is the depot, and i, j are candidate customers for pairing.

Variable Neighborhood Descent (VND)

The VND algorithm aims to find the local optimal solution $s^* \in S$ that minimizes the objective function $f : S \to \mathbb{R}$ over the feasible solution space S. It consists on the steps:

- **Initialization:** Start with an initial solution $s \in S$ and set $k = 1$.
- **Main Loop:** While $k \leq k_{\text{max}}$:

  Perform a local search in neighborhood $\mathcal{N}_k(s)$ to find the best neighboring solution s'.

  If $f(s') < f(s)$, update $s \leftarrow s'$ and reset $k = 1$; otherwise, increment k by 1.
- **Stopping Condition:** Stop when the time limit is reached.

Neighborhoods Used

- **Swap Between Routes:** Swaps customers between two routes if feasible.
- **Or-Opt Within Route:** Moves small segments within a route to improve structure.
- **Relocate Between Routes:** Moves a customer from one route to another.
- **2-Opt Within Route:** Reverses a segment within a route to shorten it.
- **2-Opt Across Routes:** Exchanges segments between two routes.
- **Merge Routes:** Merges two routes into one if feasible.

Constructive Results

Instance Data				Constructive				Constructive + Metaheuristic				
Instance	n	LB_K	LB_D	K	D	GAP_K	GAP_D	K	D	GAP_K	GAP_D	t [s]
VRPTW1	25	3	133.286	3	257.922	0.000	1.606	3	191.814	0.000	0.439	2.970
VRPTW2	25	1	184.187	2	328.895	1.000	1.319	1	215.543	1.000	0.170	3.648
VRPTW3	25	2	399.722	8	634.907	3.000	1.294	2	618.330	3.000	0.547	4.077
VRPTW4	25	1	314.469	2	745.996	1.000	1.695	1	524.593	1.000	0.668	3.972
VRPTW5	25	3	162.712	5	567.906	0.667	2.183	3	462.156	0.333	1.840	3.639
VRPTW6	25	1	189.481	2	681.966	1.000	2.823	2	467.953	1.000	1.470	4.015
VRPTW7	50	5	252.397	5	493.598	0.000	1.523	5	363.247	0.000	0.439	5.467
VRPTW8	50	2	306.019	2	515.023	0.000	0.950	2	444.961	0.000	0.454	6.093
VRPTW9	50	4	623.861	13	1256.124	2.250	2.010	4	1054.750	2.000	0.691	8.521
VRPTW10	50	1	512.907	3	1435.358	2.000	2.439	1	878.746	2.000	0.713	7.666
VRPTW11	50	5	299.824	9	1060.250	0.800	2.358	5	970.500	0.800	2.237	7.214
VRPTW12	50	1	252.250	3	1391.370	2.000	3.407	1	889.784	2.000	2.527	7.050
VRPTW13	100	10	549.659	10	889.047	0.000	1.131	10	828.937	0.000	0.508	15.795
VRPTW14	100	3	529.193	3	779.902	0.000	0.584	3	591.557	0.000	0.118	25.784
VRPTW15	100	8	872.746	22	1962.402	1.750	2.490	8	1660.249	1.500	0.902	25.515
VRPTW16	100	2	684.174	5	2208.197	1.500	2.927	2	1368.100	1.500	1.000	35.997
VRPTW17	100	9	665.344	18	2185.075	1.000	2.874	9	1749.761	0.889	1.630	25.290
VRPTW18	100	2	628.503	5	2938.973	1.500	4.211	2	1462.703	1.500	1.327	38.964
					Mean:	**1.081**	**2.101**		Mean:	**1.029**	**0.982**	

Figure: Results for the 18 instances using the constructive initial solution. The improvement percentage in K (number of routes) was 4.85%, while the improvement percentage in D (total distance) reached 53.25%.

GRASP Results

Instance Data				Constructive				GRASP + Metaheuristic				
Instance	n	LB_K	LB_D	K	D	GAP_K	GAP_D	K	D	GAP_K	GAP_D	t[s]
VRPTW1	25	3	133.286	3	257.922	0.000	1.606	3	191.814	0.000	0.439	3.081
VRPTW2	25	1	184.187	2	328.895	1.000	1.319	2	215.543	1.000	0.170	3.708
VRPTW3	25	2	399.722	8	634.907	3.000	1.294	8	618.330	3.000	0.547	4.218
VRPTW4	25	1	314.469	2	745.996	1.000	1.695	2	539.306	1.000	0.715	4.022
VRPTW5	25	3	162.712	5	567.906	0.667	2.183	4	462.156	0.333	1.840	3.968
VRPTW6	25	1	189.481	2	681.966	1.000	2.823	2	483.759	1.000	1.553	3.797
VRPTW7	50	5	252.397	5	493.598	0.000	1.523	5	363.247	0.000	0.439	6.662
VRPTW8	50	2	306.019	2	515.023	0.000	0.950	2	444.961	0.000	0.454	8.458
VRPTW9	50	4	623.861	13	1256.124	2.250	2.010	13	1061.213	2.250	0.701	7.347
VRPTW10	50	1	512.907	3	1435.358	2.000	2.439	3	884.535	2.000	0.725	7.169
VRPTW11	50	5	299.824	9	1060.250	0.800	2.358	9	965.129	0.800	2.219	7.197
VRPTW12	50	1	252.250	3	1391.370	2.000	3.407	3	892.827	2.000	2.539	8.155
VRPTW13	100	10	549.659	10	889.047	0.000	1.131	10	828.937	0.000	0.508	23.646
VRPTW14	100	3	529.193	3	779.902	0.000	0.584	3	591.557	0.000	0.118	44.652
VRPTW15	100	8	872.746	22	1962.402	1.750	2.490	21	1670.825	1.625	0.914	24.313
VRPTW16	100	2	684.174	5	2208.197	1.500	2.927	7	1178.255	2.500	0.722	36.657
VRPTW17	100	9	665.344	18	2185.075	1.000	2.874	17	1741.908	0.889	1.618	25.063
VRPTW18	100	2	628.503	5	2938.973	1.500	4.211	6	1408.347	2.000	1.241	28.340
					Mean:	**1.081**	**2.101**		Mean:	**1.133**	**0.970**	

Figure: Results for the 18 instances using the GRASP initial solution. The new solution worsened in K (number of routes) by 4.87%, while the improvement percentage in D (total distance) reached 53.83%.

ACO Results

Instance Data				Constructive				Constructive + Metaheuristic				
Instance	**n**	**LB_K**	**LB_D**	**K**	**D**	**GAP_K**	**GAP_D**	**K**	**D**	**GAP_K**	**GAP_D**	**t [s]**
VRPTW1	25	3	133.286	3	257.922	0.000	1.606	3	191.814	0.000	0.439	2.970
VRPTW2	25	1	184.187	2	328.895	1.000	1.319	1	215.543	1.000	0.170	3.648
VRPTW3	25	2	399.722	8	634.907	3.000	1.294	2	618.330	3.000	0.547	4.077
VRPTW4	25	1	314.469	2	745.996	1.000	1.695	1	524.593	1.000	0.668	3.972
VRPTW5	25	3	162.712	5	567.906	0.667	2.183	3	462.156	0.333	1.840	3.639
VRPTW6	25	1	189.481	2	681.966	1.000	2.823	2	467.953	1.000	1.470	4.015
VRPTW7	50	5	252.397	5	493.598	0.000	1.523	5	363.247	0.000	0.439	5.467
VRPTW8	50	2	306.019	2	515.023	0.000	0.950	2	444.961	0.000	0.454	6.093
VRPTW9	50	4	623.861	13	1256.124	2.250	2.010	4	1054.750	2.000	0.691	8.521
VRPTW10	50	1	512.907	3	1435.358	2.000	2.439	1	878.746	2.000	0.713	7.666
VRPTW11	50	5	299.824	9	1060.250	0.800	2.358	5	970.500	0.800	2.237	7.214
VRPTW12	50	1	252.250	3	1391.370	2.000	3.407	1	889.784	2.000	2.527	7.050
VRPTW13	100	10	549.659	10	889.047	0.000	1.131	10	828.937	0.000	0.508	15.795
VRPTW14	100	3	529.193	3	779.902	0.000	0.584	3	591.557	0.000	0.118	25.784
VRPTW15	100	8	872.746	22	1962.402	1.750	2.490	8	1660.249	1.500	0.902	25.515
VRPTW16	100	2	684.174	5	2208.197	1.500	2.927	2	1368.100	1.500	1.000	35.997
VRPTW17	100	9	665.344	18	2185.075	1.000	2.874	9	1749.761	0.889	1.630	25.290
VRPTW18	100	2	628.503	5	2938.973	1.500	4.211	2	1462.703	1.500	1.327	38.964
					Mean:	**1.081**	**2.101**		Mean:	**1.029**	**0.982**	

Figure: Results for the 18 instances using the ACO initial solution. The improvement percentage in K (number of routes) was 6.06%, while the improvement percentage in D (total distance) reached 53.35%.

Final Results

Instance Data				Best Metaheuristic Results				
Instance	n	LB_K	LB_D	K	D	GAP_K	GAP_D	t [s]
VRPTW1	25	3	133.286	3	191.814	0.000	0.439	2.970
VRPTW2	25	1	184.187	2	215.543	1.000	0.170	3.648
VRPTW3	25	2	399.722	8	618.330	3.000	0.547	4.077
VRPTW4	25	1	314.469	2	524.593	1.000	0.668	3.972
VRPTW5	25	3	162.712	4	462.156	0.333	1.840	3.639
VRPTW6	25	1	189.481	2	467.953	1.000	1.470	4.015
VRPTW7	50	5	252.397	5	363.247	0.000	0.439	5.467
VRPTW8	50	2	306.019	2	444.961	0.000	0.454	6.093
VRPTW9	50	4	623.861	12	1054.141	2.000	0.690	8.555
VRPTW10	50	1	512.907	2	1051.417	1.000	1.050	7.846
VRPTW11	50	5	299.824	9	965.129	0.800	2.219	7.197
VRPTW12	50	1	252.250	3	854.520	2.000	2.388	7.820
VRPTW13	100	10	549.659	10	828.937	0.000	0.508	16.871
VRPTW14	100	3	529.193	3	591.557	0.000	0.118	32.595
VRPTW15	100	8	872.746	19	1683.094	1.375	0.929	32.465
VRPTW16	100	2	684.174	5	1368.100	1.500	1.000	35.997
VRPTW17	100	9	665.344	16	1700.925	0.778	1.556	29.612
VRPTW18	100	2	628.503	5	1462.703	1.500	1.327	38.964
					Mean:	**0.9603**	**0.9895**	

Figure: Best results for each instance were selected based on the initial solution method. The obtained GAPs were $GAP_K = 96\%$ and $GAP_D = 99\%$ compared to the lower bounds from [4].

Simulated Annealing Parameter Comparison

This experiment analyzes the impact of Simulated Annealing parameters on solution quality without the last VND. Four variations were tested, keeping Tabu Search parameters constant.

Algorithm	**Parameter**	**Default**	**Var 1**	**Var 2**	**Var 3**	**Var 4**
Simulated Annealing	T_0	300	500	200	100	50
Simulated Annealing	α	0.95	0.90	0.85	0.80	0.75
Tabu Search	τ	10	10	10	10	10
Tabu Search	α	20	20	20	20	20
Tabu Search	β	200	200	200	200	200

Table: Simulated Annealing Parameter Variations with Constant Tabu Search Parameters.

Simulated Annealing Results in ACO

The default configuration ($T_0 = 300$, $\alpha = 0.95$) provided the best performance in balancing route and distance GAPs. Variations with lower T_0 and α converged faster but at the cost of solution quality.

Variation	Parameters	Route GAP	Distance GAP
Default	$T_0 = 300, \alpha = 0.95$	1.596	0.901
Var 1	$T_0 = 500, \alpha = 0.90$	1.751	0.912
Var 2	$T_0 = 200, \alpha = 0.85$	1.911	0.920
Var 3	$T_0 = 100, \alpha = 0.80$	1.886	0.923
Var 4	$T_0 = 50, \alpha = 0.75$	1.697	0.916

Table: Route and Distance GAPs for ACO with Simulated Annealing Variations.

Tabu Search Parameter Comparison

This experiment explores the effect of Tabu Search parameters, keeping Simulated Annealing parameters constant. Four variations were evaluated.

Algorithm	**Parameter**	**Default**	**Var 1**	**Var 2**	**Var 3**	**Var 4**
Simulated Annealing	T_0	300	300	300	300	300
Simulated Annealing	α	0.95	0.95	0.95	0.95	0.95
Tabu Search	τ	10	8	16	20	12
Tabu Search	α	20	25	35	40	30
Tabu Search	β	200	250	350	400	300

Table: Tabu Search Parameter Variations with Constant Simulated Annealing Parameters.

Tabu Search Results in ACO

The default configuration ($\tau = 10$, $\alpha = 20$, $\beta = 200$) provided the best results, balancing route and distance GAPs. Higher values led to reduced flexibility and suboptimal performance in route organization.

Variation	Parameters	Route GAP	Distance GAP
Default	$\tau = 10, \alpha = 20, \beta = 200$	1.596	0.901
Var 1	$\tau = 8, \alpha = 25, \beta = 250$	1.824	0.918
Var 2	$\tau = 16, \alpha = 35, \beta = 350$	1.647	0.908
Var 3	$\tau = 20, \alpha = 40, \beta = 400$	1.614	0.909
Var 4	$\tau = 12, \alpha = 30, \beta = 300$	1.790	0.909

Table: Route and Distance GAPs for ACO with Tabu Search Variations.

Conclusions

The exhaustive definition of neighborhoods in VND, focusing on efficient node relocation and organization within routes, enhances the exploration of solution configurations, significantly improving algorithm performance. Robust neighborhoods ensure better search quality, reducing distance costs and optimizing route utilization.

The use of creation and destruction operators in ALNS minimizes vehicle counts by optimally reorganizing route segments. This complements Simulated Annealing and Tabu Search, improving escape from local minima and expanding solution exploration toward the global optimum.

Insights from Experimental Observations

High acceptance probabilities in Simulated Annealing can lead to extremely poor solutions when the algorithm fails to converge properly without applying VND refinement at the end.

Tabu Search occasionally prioritizes minimizing distance, prematurely closing time windows and leading to suboptimal solutions by failing to balance route feasibility with cost minimization.

The addition of VND after the main algorithms significantly enhances final solutions, demonstrating its potential as a crucial refinement step to achieve better results.

ALNS shows inefficiency in creation steps, as a disproportionate amount of computational effort is spent rebuilding routes compared to destruction phases.

Parameter Comparison Insights

Default parameters in Simulated Annealing ($T_0 = 300, \alpha = 0.95$) provided the best balance between route and distance GAPs. Variations with lower T_0 and α showed faster convergence but poorer solution quality.

In Tabu Search, moderate tabu tenure ($\tau = 10$) and weight values ($\alpha = 20, \beta = 200$) achieved the best performance. Higher values reduced flexibility, negatively impacting route optimization.

The sensitivity of the algorithms to parameter tuning highlights the importance of careful calibration to ensure stable and effective metaheuristic performance across diverse instances.

References

[1] J. d. C. Peralta-Abarca and E. Segura-Artavia, "Aplicaciones de recocido simulado en problemas de optimización combinatoria," *Inventio*, vol. 11, no. 25, pp. 23–28, 2015.

[2] N. Yusof, S. Hashim, N. Abdul Hamid, and S. N. Zainudin, "A tabu search approach for solving university course timetabling problems," *Procedia-Social and Behavioral Sciences*, vol. 176, pp. 576–582, 2015.

[3] D. Pisinger and S. Ropke, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, 2007.

[4] J. C. R. Agudelo, *Métodos constructivos y aleatorizados*, Curso: Heurística – CM0439, Universidad EAFIT, Medellín, Colombia, Jul. 15, 2024.