University of Pittsburgh

PITT | SWANSON ENGINEERING

INDUSTRIAL ENGINEERING
SUMMER 2021

TRANSPORTATION PLANNING/ IE 2045
BY
PROF. NATASA VIDIC

Research Paper review

SUBMITTED BY

SABBELLA PRASANNA
4450873
prs98@pitt.edu

Background

Problem Statement

Mathematical modeling

Methodology

Evaluation of the algorithm

Conclusion

Background

Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

Yun-qi Han[1], Jun-qing Li[1,2], Zhengmin Liu[3], Chuang Liu[2] and Jie Tian[1]

- A special case of a Vehicle routing problem is to transport needed goods using drones because of the inaccessible landform.
- This is a Vehicle routing problem with multiple objectives and time window and the vehicles also include the drones for vertical transportation. (MO-VRPTW-D)
- Most helpful in natural disaster scenarios such as landslides and flood relief.

Problem Statement

Problem Statement

- The authors considered different types of vehicles based on their carrying capacity and marked demands for every customers.
- Energy consumption of these vehicles including the drones is also different.
- Each truck can only carry one drone.
- Customer's location is first addressed as a 2D point on a plane and then the perpendicular height is taken into account.

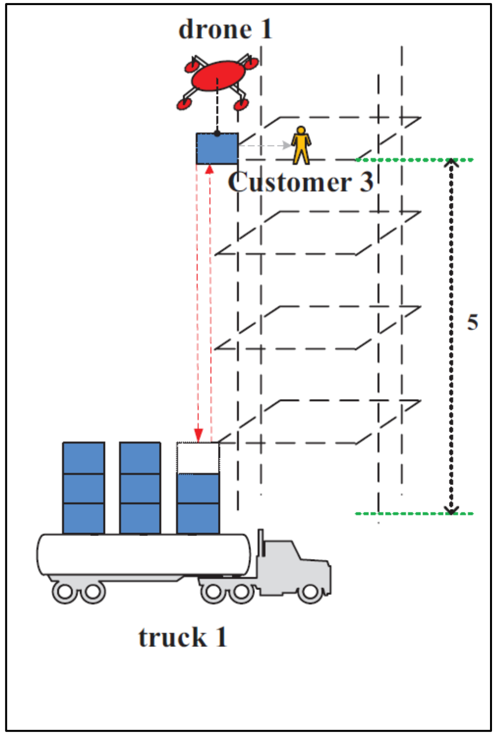
Problem statement is to find out the best possible combinations of these trucks and drones such that the total energy consumption ($\propto$ travel distance) is minimum without violating the time window margins.

Mathematical Modeling

Objective Function

$$\min \alpha \underbrace{\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{v} x_{ijk}\, wt_k\, t_{ij}}_{\text{weighted sum of the total energy consumption of the trucks}} + \beta \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{v} x_{0jk}\, 2tp_i \frac{1}{yd_k} wd_k}_{\text{weighted sum of the total energy consumption of the drones}} + \gamma \underbrace{\sum_{k=1}^{v} x_{0jk}}_{\text{Total number of the trucks/drones}},\ i \neq j, i, j \in N, k \in V$$

- x_{ijk} = 1 if Vehicle k travels from i to j, else 0
- yik = 1 if vertex i is served by vehicle k
- yd_k = Speed of the drone
- wt_k = Energy consumption coefficient of truck k
- wd_k = Energy consumption coefficient of drone k
- t_{ij} = Travelled distance from i to j (0 if X_{ijk}= 0)
- tp_i = Travelled height by drone at i



Constraint Functions

$$\sum_{i=1}^{n} dm_i y_{ik} \leq bm_k, \forall k \in V$$

$$\sum_{i=1}^{n} ds_i y_{ik} \leq bs_k, \forall k \in V$$

Customer demands do not exceed the truck supply capacity

$$\sum_{k=1}^{v} y_{ik} = 1, \forall i \in C$$

$$\sum_{i=0}^{n} x_{ijk} = y_{jk}, \forall k \in V, \forall j \in C$$

$$\sum_{j=0}^{n} x_{ijk} = y_{ik}, \forall k \in V, \forall i \in C$$

Each customer sees a truck and only one truck

$$\sum_{j=1}^{n} x_{0jk} - \sum_{i=1}^{n} x_{i0k} = 0, \forall k \in V$$

Each truck starts and ends at depot

$$s_i = 2tp_i \frac{1}{yd_k}, \forall i \in C, \forall k \in V$$

Time = (Distance/Speed); Serving duration

$$\sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk}(t_{ij} + s_i + w_i) \leq r_k, \forall k \in V$$

Maximum trip duration does not exceed the limit

$$w_0 = s_0 = 0$$

No waste of time at the depot

$$\sum_{k=1}^{v} \sum_{i=0}^{n} x_{ijk}(z_i + w_i + s_i + t_{ij}) = z_j, \forall j \in C$$

Departure time at i should be arrival time at j

$$e_i \leq z_i + w_i \leq l_i, \forall i \in C$$

$$w_i = \max\{0, e_i - z_i\}, \forall i \in C$$

Arrival time and wait time should be within the time window

$$x_{ijk} \in \{0,1\}, \forall i,j \in C, \forall k \in V$$

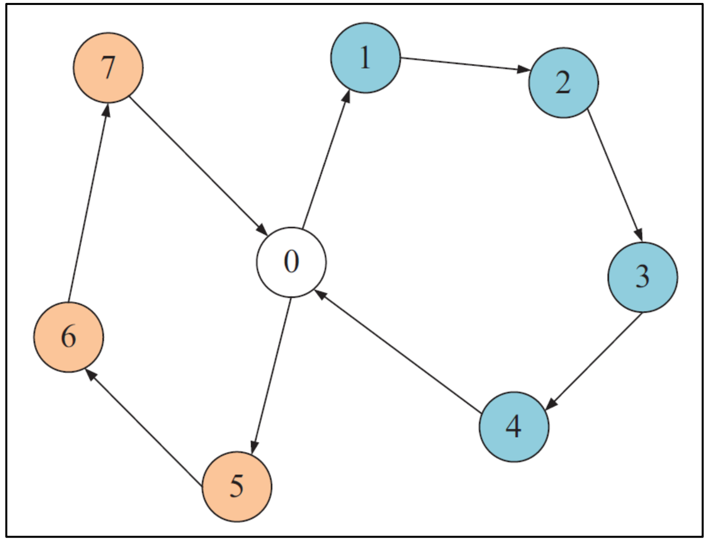
$$y_{ik} \in \{0,1\}, \forall i \in C, \forall k \in V$$

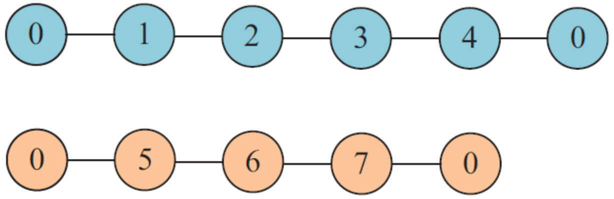
$$z_i \geq 0, \forall i \in C$$

- x_{ijk} = 1 if vehicle k travels from i to j, else 0
- y_{ik} = 1 if vertex i is served by vehicle k
- s_i = Serving duration
- yd_k = Speed of the drone
- wt_k = Energy consumption coefficient of truck k
- wd_k = Energy consumption coefficient of drone k
- t_{ij} = Travelled distance from i to j (0 if X_{ijk}= 0)
- tp_i = Travelled height by drone at i
- z_i = Arrival time of truck k at customer i
- [e_i l_i] = Time limit window at vertex i
  e_0 = 0 and l_0 = 0
- [e_i l_i] = Time limit window at vertex I
- wi = wait time if $z_i < e_i$

Critique points

- Though the energy consumption coefficient of the trucks (wt_k) and the drones (wd_k) mostly depend on their engine technology and its delivered speed, we are only limited to optimize by choosing different vehicles/drones and by selecting different routes.
- One drone on one truck constraint is missing.
- No where in the paper did the author mention what trucks are chosen with what drones.
- Bias: The weights (α, β & Γ) in the objective function are unknown.





Methodology

Methodology overview

- The **artificial bee colony (ABC) algorithm** was first proposed by Author Karaboga to solve VRP.
- Author of our paper has modified this ABC algorithm by inducing **an Enhanced employed bee strategy and scout bee strategy**
- Also induced repair strategy and Initialization strategy
- This new modified strategy is named as **Improved Artificial Bee Colony (IABC) algorithm**.

The IABC algorithm

Input: the datas of customers, trucks and drones
Output: the best solution X^*
1 generate P_s initial solutions X by the strategy in sub-section 3.4
2 **employed bee phase**
for each solution x **do**
3 generate a new solution x^* by the strategy in sub-section 3.5 for solution x
4 **if** x^* is better than x
5 replace x with x^*
6 update the best solution that has been found so far to x^*
7 update the local best to x
8 **end**
9 **end**
10 **onlooker bee phase**
for each solution x **do**
11 randomly select a solution x_r
12 implement the Employed bee phase on the best solution between x and x_r
13 update the best solution X^*
14 **end**
15 **scout bee phase**
implement the Scout bee strategy for the best solution X^*
16 update the iterative time T of the IABC algorithm
17 update iteration times without improvement I_x for each solution x
18 if $I_{x'}$ exceeds the maximum no- improvement time L_n
19 implement the Scout bee strategy for the solution x'
20 **if** T not exceeds the $T_{\max}$
21 go to step 2

The **Repair Strategy** replaces infeasible solution with feasible ones.

- The traditional employed bee strategy chooses customers randomly of certain route, then re-arrange these customers to other routes.
- Easy to accomplish but hard to reach to an improved solution.
- Thus, use **Enhanced employed bee strategy** which chooses customers not randomly but based on **Push Forward Insertion Heuristic (PFIH)**

- The **Scout bee strategy** makes large changes to the solution.
- Executes a local search to insert some customers of solution
- Replace the truck of this solution with another new one.

Evaluation

Effectiveness of the Enhanced employed bee strategy / using deviations

Instance	Random	Enhanced (PFIH)
Inst1	**0.00**	3.65
Inst2	6.25	**0.00**
Inst3	**0.00**	4.57
Inst4	**0.00**	2.75
Inst5	**0.00**	1.95
Inst6	1.23	**0.00**
Inst7	2.44	**0.00**
Inst8	**0.00**	3.32
Inst9	8.12	**0.00**
Inst10	**0.00**	1.46
Inst11	7.31	**0.00**
Inst12	0.88	**0.00**
Inst13	11.35	**0.00**
Inst14	**0.00**	0.10
Inst15	**0.00**	2.34
Inst16	5.08	**0.00**
Inst17	8.51	**0.00**
Inst18	3.35	**0.00**
Inst19	2.14	**0.00**
Inst20	1.05	**0.00**
Inst21	5.45	**0.00**
Inst22	**0.00**	0.92
Inst23	3.45	**0.00**
Inst24	6.36	**0.00**
Inst25	**0.00**	1.28
Inst26	1.52	**0.00**
Inst27	7.48	**0.00**
Inst28	**0.00**	0.15
Inst29	0.18	**0.00**
Inst30	7.08	**0.00**
Inst31	**0.00**	8.35

Instance	Random	Enhanced (PFIH)
Inst32	1.04	**0.00**
Inst33	3.48	**0.00**
Inst34	2.55	**0.00**
Inst35	1.33	**0.00**
Inst36	1.14	**0.00**
Inst37	**0.00**	3.03
Inst38	1.21	**0.00**
Inst39	**0.00**	1.35
Inst40	**0.00**	3.23
Inst41	0.44	**0.00**
Inst42	**0.00**	2.47
Inst43	**0.00**	2.88
Inst44	3.29	**0.00**
Inst45	**0.00**	1.75
Inst46	**0.00**	0.63
Inst47	1.99	**0.00**
Inst48	4.06	**0.00**
Inst49	3.18	**0.00**
Inst50	**0.00**	1.66
Inst51	2.20	**0.00**
Inst52	5.25	**0.00**
Inst53	1.36	**0.00**
Inst54	2.52	**0.00**
Inst55	2.31	**0.00**

Total best	20	35
Mean deviation	2.3	0.87

- Mean deviation
- Total best
- ANOVA hypothesis test

Effectiveness of the Scout bee strategy / using deviations

Instance	Random	Enhanced (PFIH)
Inst1	0.29	**0.00**
Inst2	10.22	**0.00**
Inst3	0.53	**0.00**
Inst4	**0.00**	2.32
Inst5	2.20	**0.00**
Inst6	0.25	0.00
Inst7	**0.00**	2.42
Inst8	2.16	**0.00**
Inst9	1.54	**0.00**
Inst10	**0.00**	4.34
Inst11	**0.00**	5.79
Inst12	6.98	**0.00**
Inst13	17.64	**0.00**
Inst14	**0.00**	5.00
Inst15	**0.00**	1.02
Inst16	**0.00**	2.99
Inst17	14.80	**0.00**
Inst18	0.68	**0.00**
Inst19	0.56	**0.00**
Inst20	0.10	**0.00**
Inst21	5.46	**0.00**
Inst22	6.08	**0.00**
Inst23	3.87	**0.00**
Inst24	**0.00**	2.25
Inst25	0.06	**0.00**
Inst26	1.21	**0.00**
Inst27	2.84	**0.00**
Inst28	**0.00**	0.78
Inst29	1.07	**0.00**
Inst30	0.68	**0.00**
Inst31	**0.00**	0.85
Inst32	1.69	**0.00**
Inst33	4.78	**0.00**

Instance	Random	Enhanced (PFIH)
Inst34	**0.00**	0.19
Inst35	0.90	**0.00**
Inst36	**0.00**	10.45
Inst37	**0.00**	4.10
Inst38	3.46	**0.00**
Inst39	**0.00**	3.79
Inst40	5.51	**0.00**
Inst41	**0.00**	1.32
Inst42	1.78	**0.00**
Inst43	3.16	**0.00**
Inst44	2.95	**0.00**
Inst45	2.64	**0.00**
Inst46	1.59	**0.00**
Inst47	2.33	**0.00**
Inst48	3.04	**0.00**
Inst49	0.77	**0.00**
Inst50	2.17	**0.00**
Inst51	**0.00**	2.20
Inst52	**0.00**	7.47
Inst53	6.72	**0.00**
Inst54	9.65	**0.00**
Inst55	**0.00**	2.62

Total best	18	37
Mean deviation	2.41	1.09

- Mean deviation
- Total best
- ANOVA hypothesis test

Results and Conclusion

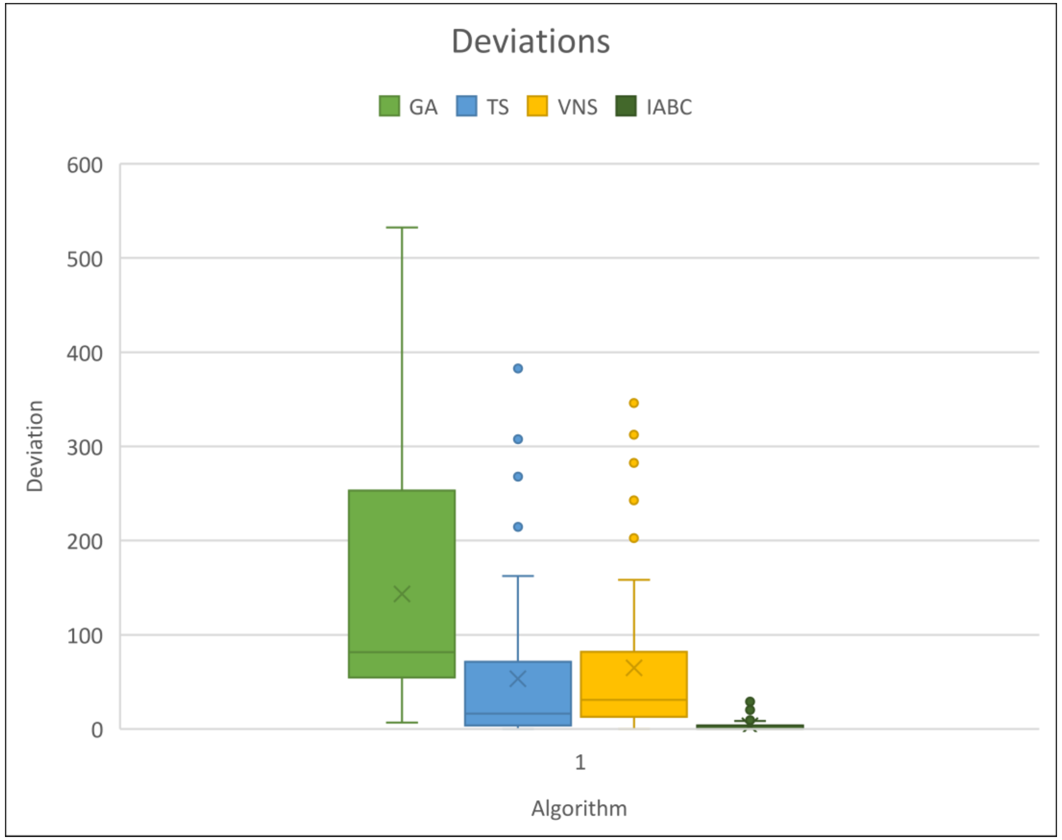
Algorithm comparisons

- The author compares the IABC three different algorithms.
- **Tabu Search (TS), Genetic Algorithm (GA) and Variable Neighborhood Search Algorithm (VNS)**
- These three algorithms are famously endorsed by many researches as the top algorithms for solving VRP.

Algorithm comparisons

	Objective value					Deviation			
Instances	Best	GA	TS	VNS	IABC	GA	TS	VNS	IABC
Inst1	1038.78	1571.34	1306.84	1351.27	1038.78	532.56	268.06	312.49	0
Inst2	1001.63	1275.01	1163.9	1125.35	1001.63	273.38	162.27	123.72	0
Inst3	950.11	1134.35	950.11	980.92	966.26	184.24	0	30.81	16.15
Inst4	956.58	1217.11	981.48	1011.2	956.58	260.53	24.9	54.62	0
Inst5	1010.41	1313.63	1108.59	1112.75	1010.41	303.22	98.18	102.34	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Inst51	544.7	627.43	564.02	569.84	544.7	82.73	19.32	25.14	0
Inst52	479.69	511.9	485.82	479.69	483.92	32.21	6.13	0	4.23
Inst53	826.79	912.19	844.77	826.79	842.6	85.4	17.98	0	15.81
Inst54	626.18	658.18	627.59	626.18	631.6	32	1.41	0	5.42
Inst55	565.48	707.02	589.38	688.58	565.48	141.54	23.9	123.1	0
Mean		931.25	841.23	852.65	791.37				
Total best		0	09	09	38				

- The author compares the IABC with TS, GA & VNS algorithms.
- IABC holds the greatest number of best solutions (=38) (70%)
- TS & VNS are close.
- GA is the worst performer.



Critique points

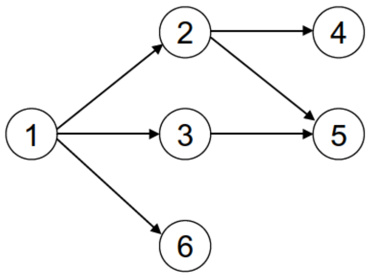
- The author has compared an improved version of ABC with raw versions of TS, GA and VNS which is not a fair comparison.
- As mentioned previously that the energy coefficients mostly depends on the engine speed/performance the author admits that constraints such as traffic intensity, accident situations, road/terrain and weather conditions are necessary.
- Modifications to algorithm contributed well for the betterment of the solution.

Background

An Efficient Genetic Algorithm for Large Scale Vehicle Routing Problem Subject to Precedence Constraints

Noraini Mohd Razali[a*]

- Vehicle routing problem with only one vehicle and precedence constraints is Travelling Salesman Problem (TSP) with precedence constraints.
- The special case of this problem is that the vehicle need not return to the depot.
- Real life vehicle routing problems are usually so large that exact methods such as TS, GA cannot be used to solve them as many heuristic problems do not seek for a global minima but a local one.
- Hence use a metaheuristic approach.



Example of directed graph

Problem Statement

Problem Statement

- The authors considers one vehicle available at the depot and a set of n customers C = {0,1 … n} and the vehicle need not report back to the depot.

- Open Vehicle Routing Problem with Precedence Constraints. (VRP-PC)

- **Problem/Goal statement:** To find the most optimal route of delivery by the vehicle without violating the precedence constraints

Mathematical Modeling

Objective function

$$\text{Minimise} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \frac{1}{(n-1)} c_{ij} \left(x_{ij}^{p} + x_{ij}^{q} \right)$$

- n = Number of nodes
- $n-1$ = Number of nodes excluding the depot
- C_{ij} = Distance to cover in transporting from i to j
- x_{ij}^{p} = Quantity of commodity p from i to j
- x_{ij}^{q} = Quantity of commodity q from i to j
- x_{ij} = 1, if j is visited immediately after i, else 0

➢ The authors considers two different commodities p and q and not much is known about them.

➢ No cost matrix. Assuming cost associated (t_{ij}) is in direct proportion to the distance matrix.

Problem Statement

$$\sum_{j=1}^{n} x_{ij}^{p} - \sum_{j=1}^{n} x_{ji}^{p} = \begin{cases} n-1 & \text{for } i = s, \\ -1 & \text{otherwise,} \end{cases}$$

Flow of commodity p is feasible (No backflow in the route)

$$\sum_{j=1}^{n} x_{ij}^{p} - \sum_{j=1}^{n} x_{ji}^{p} = \begin{cases} -(n-1) & \text{for } i = s, \\ +1 & \text{otherwise,} \end{cases}$$

Flow of commodity q is feasible (No backflow in the route)

$$\sum_{j=1}^{n} \left(x_{ij}^{p} + x_{ij}^{q} \right) = n-1 \quad \forall i,$$

Makes sure no node is left unvisited

$$x_{ij}^{p} + x_{ij}^{q} = (n-1) x_{ij} \quad \forall i \text{ and } j,$$

If x_{ij} = 1 then sum of commodities between i and j is (n-1)

$$\sum_{j=1}^{n} x_{uj}^{p} - \sum_{j=1}^{n} x_{vj}^{p} \geq 1 \quad \text{for } (v_u \to v_v)(v_v \neq s)$$

For precedence relationships between vertices

$$x_{ij}^{p} \geq 0 \quad \forall i \text{ and } j,$$

$$x_{ij}^{q} \geq 0 \quad \forall i \text{ and } j,$$

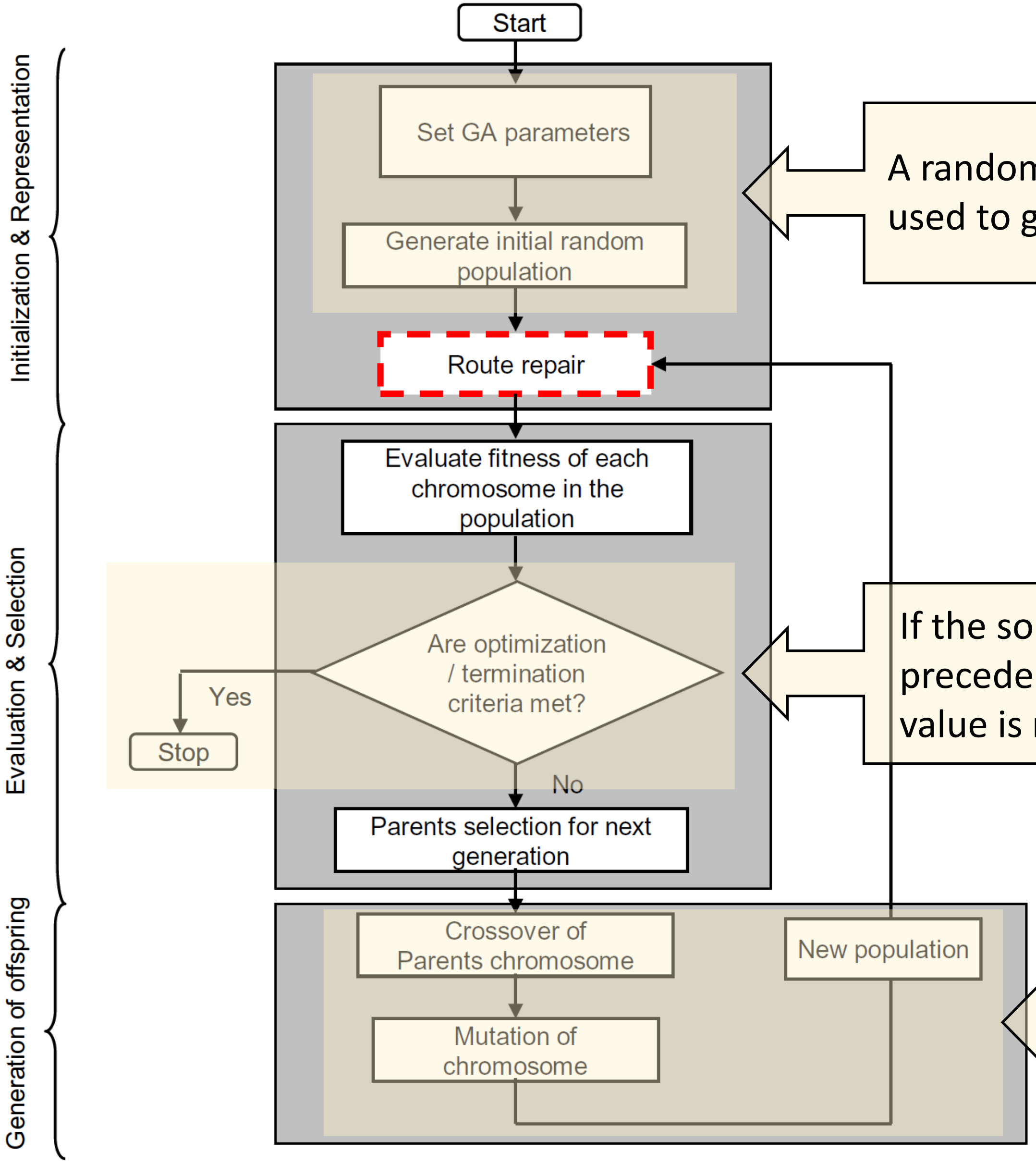
$$x_{ij} \in \{0,1\} \quad \forall i \text{ and } j.$$

Methodology

Methodology overview

- Genetic Algorithms are inspired by the theory of natural selection by Charles Darwin.
- A population of **individuals** are developed by means of a **chromosome** crossover **(reproduction)** and population with a lower **fitness value** are removed in every iteration.
- The author compares individuals to **nodes**, chromosomes to **solutions**, reproduction to the **repair strategy** and fitness value to the **objective function value**.
- The author names this new algorithm **GAnew** as she introduces a new repair strategy.
- Repair strategy in GAnew algorithm was inspired by a similar type of strategy used by Author Moon (**GAold**).
- ‘earliest position’ based topological sorting & ‘priority’ based topological sorting.

GAnew algorithm



A random permutation method is used to generate initial solution.

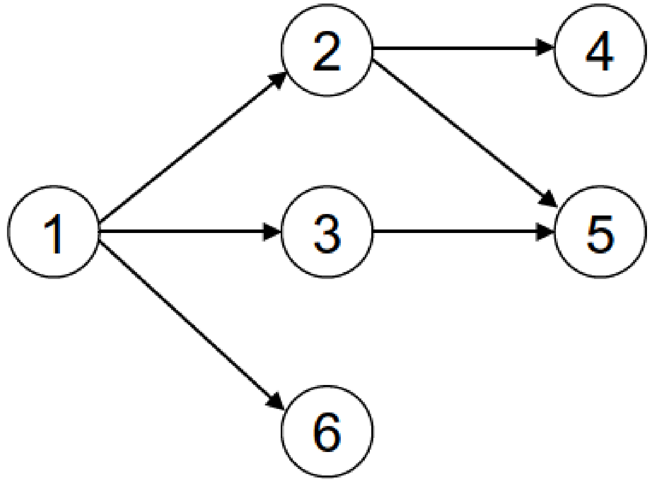
If the solution meets the precedence constraints fitness value is noted.

If the solution does not meet the precedence constraints, then crossover takes place

Let us consider a randomly generated solution route as { 4 1 3 6 5 2 }

Violation: 4 has to be preceded by 2 and 5 has to be preceded by 2 & 3.

available set	*updated sequence*
1	[1]
2, 3, 6	[1 3]
2, 6	[1 3 6]
2	[1 3 6 2]
4, 5	[1 3 6 2 4]
5	[1 3 6 2 4 5]



{1 3 6 2 4 5} is a feasible solution and fitness value is recorded.

Evaluation

Evaluation

- The author compares GAnew and GAold algorithms based on two setups.
- 51 locations (customers), 71 precedence constraints and 100 locations (customers), 141 precedence constraints.
- Number of solution possibilities with 51 locations is 51 ! = 1.5511188e+66.
- Author considers popsize = 500,1000 possibilities, Probability of crossover (P_c) = 0.6,0.9 and Probability of mutation (p_m) = 0.001, 0.2
- 8 experiments to be done.

Evaluation

51 locations (customers), 71 precedence constraints.

Experiment #	Popsize	P_c	P_m	Gen #	best
1	-1	-1	-1	96	209
2	+1	-1	-1	192	204
3	-1	+1	-1	112	224
4	+1	+1	-1	157	194
5	-1	-1	+1	122	209
6	+1	-1	+1	74	218
7	-1	+1	+1	109	219
8	**+1**	**+1**	**+1**	**193**	**184**

GAnew algorithm

Experiment #	Popsize	P_c	P_m	Gen #	best
1	-1	-1	-1	150	268
2	**+1**	**-1**	**-1**	**163**	**256**
3	-1	+1	-1	83	277
4	+1	+1	-1	72	265
5	-1	-1	+1	101	272
6	+1	-1	+1	50	282
7	-1	+1	+1	51	279
8	+1	+1	+1	95	270

GAold algorithm

Results and Conclusion

GAnew vs GAold

Summary of results for all tests using GAnew and GAold algorithm

Test Problem	GAnew			GAold		
	Gen#	Best	Convergence time (sec)	Gen#	Best	Convergence time (sec)
51 tasks & 71 PC	193	184	650	163	256	1109
100 tasks & 141 PC	174	441	928	381	559	1240

Critique points

- Author did not mention how she chose popsize = 500,1000 possibilities
- Probability of crossover (P_c) and mutation (p_m) are very critical and in the analysis, they are just fixed.