Inteligencia Artificial

Informe Final: Truck and trailer routing problem

[Pedro Herrera]

14 de diciembre de 2022

Evaluación

Mejoras 1ra Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (15 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
Nota Final (100):	_____

Resumen

Este texto centrará sus esfuerzos en explicar detalladamente que es el Truck and trailer routing problem (**TTRP**) el cual es una variante del conocido problema VRP con la diferencia que en el TTRP se contemplan 3 tipos de rutas **PTR**, **PVR** Y **CVR**, además de repasar la historia de este problema, los avances de los investigadores en la búsqueda de la solución de este problema, los modelos matemáticos utilizados por los investigadores y las principales técnicas utilizadas, todo esto en base a una serie de investigaciones y papers científicos encontrados por la web

1. Introducción

Este informe tiene como objetivo mostrar y explicar de forma detallada la investigación del problema Truck and trailer routing problem (**TTRP**). Primeramente en la sección 2 (**Definición del problema**), se explicará en qué consiste el problema, su objetivo primordial, sus variables y restricciones. Para luego en la sección 3 (**Estado del arte**) mencionar cómo está la investigación actualmente donde se abordarán preguntas del tipo ¿cuándo surge?, ¿qué métodos se han usado para resolverlo?, ¿cuáles son los mejores algoritmos que se han creado hasta la fecha? entre otras. En la sección 4 (**Modelo matemático**) se analizarán los modelos matemáticos obtenidos por los investigadores. En la sección 5 (**Representación**) se mostraran y explicaran las estructuras de datos utilizadas además de mostrar como se entregara la solución.En la sección 6 (**Descripción del algoritmo**) se explicara como se planteo el algoritmo, que técnicas se utilizaron y como se implementaron. En la sección 7 (**Experimentos**) comentaremos los experimentos realizados al algoritmo además de definir el entorno de experimentación y los parámetros de experimentación. En la sección 8 (**Resultados**) se hablara de los logros obtenidos en la experimentación y finalmente en la sección 9 (**Conclusiones**) se comentaran las conclusiones obtenidas en el desarrollo del informe.

El TTRP es una variante del conocido problema de enrutamiento de vehículos VRP ***" que es uno de los problemas de optimización combinatoria más estudiados en las últimas décadas"***[13] con el objetivo de servir a todos los clientes minimizando la distancia total recorrida.

La motivación que hay para resolver este problema es la importancia que tiene este en la vida real, específicamente en la gestión de distribuciones y el área logística

2. Definición del Problema

El problema a estudiar es el **TTRP** o Truck and trailer routing problem. El problema de enrutamiento de camiones y remolques es una variante del problema de enrutamiento de vehículos, la principal diferencia es que en los problemas de enrutamiento de camiones y remolques (**TTRP**), una flota de camiones y remolques atiende a un conjunto de clientes. Algunos clientes pueden ser atendidos por un vehículo completo (es decir, un camión tirando de un remolque), mientras que otros clientes solo pueden ser atendidos por un solo camión debido a algunas limitaciones prácticas, como regulaciones gubernamentales, espacio de maniobra limitado en el sitio del cliente, condiciones de la carretera. Tales limitaciones existen en muchos escenarios del mundo real. Para ese propósito, los remolques se pueden desacoplar en ruta en los clientes donde se construyen los sub-recorridos de camiones.[14]. Aquellos clientes que solo pueden ser atendidos por un solo camión se denominan clientes de camión (**TC**); los otros clientes que pueden ser atendidos por un solo camión o un vehículo completo se denominan clientes de vehículos (**VC**).

Debido a la existencia de 2 tipos de clientes podemos derivar 3 tipos de rutas. La ruta de camión pura(**PTR**) donde el camión sirve a los clientes sin necesidad de tráiler, la ruta de vehículo pura(**PVR**) donde el camión sirve a todos los clientes de su ruta con el tráiler acoplado y la ruta de vehículo completa(**CVR**) donde el camión puede desacoplar el tráiler para atender a los clientes de difícil acceso. Además el **TTRP** considera restricciones tales como:

- No se puede servir dos veces a un mismo cliente.
- El camión debe terminar en el depósito donde comenzó su recorrido.
- En **PVR** o **CVR** la demanda de la ruta debe ser menor o igual a la capacidad total entre el camión y su tráiler.
- En **PTR** o sub-tours la demanda no debe sobrepasar la capacidad del camión.
- Un camión que desacople su tráiler y realice un sub-tour deberá volver a buscar el tráiler al finalizar el sub-tour.

Dentro del problema de **TTRP** existen variantes como son:

- Aplicaciones específicas en la industria de la leche.
- La imposibilidad de desacoplar el tráiler.
- Con o sin la posibilidad de transferir bienes entre compartimientos (vehículo y tráiler).
- El problema de enrutamiento de camiones y remolques con ventanas de tiempo (**TTRPTW**)[3] donde los clientes solo aceptan ser atendidos durante un período de tiempo específico
- Con la existencia de múltiples depósitos.

- El problema relajado de enrutamiento de camiones y remolques (**RTTRP**) [13] el cual tiene como objetivo reducir aún más el costo total de enrutamiento a través de relajar la restricción del tamaño de la flota.
- El problema generalizado de enrutamiento de camiones y remolques (**GTTRP**)[3] que es una generalización bastante compleja motivada por un escenario del mundo real. En este caso, los camiones y remolques pueden ser vehículos de recogida o vehículos de apoyo. Los vehículos de recogida se utilizan para recoger los suministros de los clientes, mientras que los vehículos de apoyo se utilizan como "depósitos móviles"que no pueden visitar a los clientes.

Como ya se mencionó anteriormente el **TTRP** es un derivado del **VRP** la diferencia es que en el problema clásico de enrutamiento de vehículos (**VRP**), queremos determinar un conjunto de rutas con un costo total mínimo para una flota de vehículos homogéneos, para suministrar bienes a clientes que tienen una demanda conocida. Mientras que en el **TTRP** cada ruta tiene que empezar y terminar en un depósito común, y la demanda de cada cliente tiene que ser satisfecha con una única entrega respetando la capacidad de los vehículos.

3. Estado del Arte

El problema clásico de enrutamiento de vehćulos (**VRP**) fue estudiado por primera vez por Dantzig y Ramser [5] con el objetivo de determinar un conjunto de rutas con un costo total mńimo para una flota de vehćulos homogéneos, para suministrar bienes a clientes que tienen una demanda conocida. A lo largo de los años, se han introducido en la literatura numerosas variantes de este **VRP** clásico, como por ejemplo el **VRP** con ventanas de tiempo (**VRPTW**), donde las rutas deben cumplir con las restricciones adicionales de que la entrega a cada cliente debe realizarse dentro de una ventana de tiempo específica. El **TTRP** fue propuesto por primera vez por Chao (2002)[2] y posteriormente estudiado por Scheuerer (2006)[12], Lin, Yu y Chou (2009)[13], Villegas(2011) y por ultimo Derigs, Pullmann y Vogel(2012)[13] . En **TTRP** se considera el uso de tráileres (una característica comúnmente descuidada en el **VRP**). Con el objetivo de servir a todos los clientes minimizando la distancia total recorrida, o costo total incurrido por la flota. El problema es más difcíl de resolver que el problema básico de generación de rutas para vehículos.

La literatura sobre las aplicaciones de **TTRP** es tan escasa que encontramos solo dos investigadores que hablaron de este problema.

- Gerdessen [7] describió dos aplicaciones del mundo real en la industria láctea holandesa.
- Hoff [6] estudió un **TTRP** del mundo real que ocurrió en una empresa láctea noruega.

Chao[2] utiliza el método de búsqueda tabú la cual fue presentada por primera vez por Glover[4] y luego esbozada por Hansen[10], Consiste en un procedimiento heurístico de mejora general, La cual ha tenido éxito en una variedad de entorno de problemas como la programación, el transporte, la distribución y el diseño de circuitos, y los gráficos. Chao[2] utilizó una restricción tabú basada en la frecuencia (**FTB**) que prohíbe algunos movimientos hacia atrás, además utilizó el concepto de desviación que se encuentra en el recocido determinista(**DA**). Al usar **FTB** junto con **DA** Chao[2] logró desarrollar una restricción tabú basada en objetivos, la cual implementa estrategias de intensificación y diversificación variando los valores de las desviaciones dentro de dos rangos diferentes. Con esto Chao[2] resolvió el **TTRP** de manera consistente, efectiva y eficiente como se observa en la siguiente tabla.

Problem	Initial (A)	Best (B)	Percentage improvement of the initial solution (%)					Percentage above the best solution (%)				
			Set 1 $(A-T_1)/A$	Set 2 $(A-T_2)/A$	Set 3 $(A-T_3)/A$	Set 4 $(A-T_4)/A$	Set 5 $(A-T_5)/A$	Set 1 $(T_1-B)/B$	Set 2 $(T_2-B)/B$	Set 3 $(T_3-B)/B$	Set 4 $(T_4-B)/B$	Set 5 $(T_5-B)/B$
1	646.02	565.02	9.28	8.46	12.54	12.54	12.54	3.73	4.67	0.00	0.00	0.00
2	739.90	658.07	9.21	10.96	11.06	10.41	10.41	2.07	0.12	0.00	0.72	0.72
3	774.78	648.74	12.55	12.99	12.12	16.27	14.20	4.44	3.92	4.95	0.00	2.46
4	943.47	856.20	8.05	9.25	8.75	8.99	9.08	1.32	0.00	0.55	0.29	0.19
5	1130.85	949.98	12.55	10.55	11.61	10.44	15.99	4.10	6.48	5.22	6.62	0.00
6	1236.69	1053.23	14.50	12.42	14.83	12.13	12.28	0.39	2.83	0.00	3.18	3.00
7	906.31	832.56	7.93	7.09	7.42	8.14	7.56	0.22	1.14	0.78	0.00	0.63
8	971.6	900.54	5.52	7.31	7.22	7.23	6.74	1.94	0.00	0.10	0.09	0.62
9	1106.66	971.62	10.18	11.46	11.29	12.20	9.61	2.3	0.84	1.04	0.00	2.95
10	1159.78	1073.50	7.42	6.79	7.44	6.59	7.15	0.02	0.7	0.00	0.91	0.31
11	1288.74	1170.17	8.01	8.59	8.92	8.59	9.20	1.31	0.67	0.31	0.67	0.00
12	1453.82	1217.01	11.22	11.02	13.51	15.33	16.29	6.05	6.29	3.32	1.14	0.00
13	1481.4	1364.50	6.66	6.50	6.61	6.55	7.89	1.33	1.51	1.39	1.46	0.00
14	1624.96	1464.20	8.12	6.50	8.50	8.76	9.89	1.97	3.77	1.54	1.26	0.00
15	1858.87	1540.25	17.14	14.98	15.82	14.63	16.93	0.00	2.61	1.60	3.02	0.26
16	1267.87	1041.36	7.67	15.75	9.61	17.87	16.01	12.41	2.58	10.05	0.00	2.26
17	1261.17	1090.46	9.35	13.54	11.33	12.78	12.41	4.84	0.00	2.55	0.87	1.30
18	1366.21	1141.36	13.90	12.58	13.76	16.46	12.02	3.06	4.64	3.23	0.00	5.31
19	969.96	854.02	11.95	8.39	7.81	7.62	8.53	0.00	4.05	4.71	4.93	3.89
20	1140.47	942.39	17.23	17.37	15.89	15.97	15.56	0.16	0.00	1.79	1.69	2.19
21	1174.43	926.47	18.20	21.11	18.59	17.77	18.91	3.69	0.00	3.20	4.24	2.79
Average			10.79	11.12	11.17	11.77	11.87	2.64	2.23	2.21	1.48	1.38

Figura 1: Comparación de soluciones

En la figura (1) los números en negrita son las mejores soluciones.

En 2006 Stephan Scheuerer[12] propuso dos nuevas heurśticas de construcción de soluciones. **T-Cluster** y **T-Sweep** las cuales son altamente modificables para incluir restricciones secundarias de la vida real. **T-Cluster** se puede considerar como un procedimiento de inserción secuencial basado en grupos, donde las rutas se construyen una por una hasta la utilización total del vehiículo.**T-Sweep** es una heurística basada en el algoritmo de barrido clásico comúnmente atribuido a Gillett y Miller[1] . Se adapta mejor a las instancias planares con un depósito ubicado en el centro. El método construye rutas factibles girando un rayo centrado en el depósito e incluyendo gradualmente a los clientes en una ruta de vehículos.

ID	T-Cluster[a]			T-Sweep[b]			Chao constr.[c,e]		Chao descent[c,e]		$c(s^{**})$
	$c(s^*)$	$q(s^*)$	T[d]	$c(s^*)$	$q(s^*)$	T[d]	$c(s^*)$	$q(s^*)$	$c(s^*)$	$q(s^*)$	
1	651.87	0.0	0.33	644.80	0.0	0.30	657.15	9.6	646.02	0.0	564.68
2	697.51	0.0	0.27	722.46	0.0	0.23	739.04	13.9	739.90	0.0	612.75
3	766.25	0.0	0.25	797.65	0.0	0.23	785.54	16.8	774.78	0.0	618.04
4	979.79	0.0	0.45	901.78	26.0	0.56	937.82	26.0	943.47	0.0	798.53
5	1037.50	0.0	0.42	1035.76	32.0	0.55	1108.87	22.9	1130.85	0.0	839.62
6	1173.11	0.0	0.41	1171.99	52.0	0.55	1174.17	32.1	1236.69	0.0	933.26
7	904.77	0.0	1.09	901.14	0.0	1.88	937.31	14.1	906.31	0.0	830.48
8	965.90	0.0	1.00	1005.99	0.0	1.63	1004.45	18.5	971.60	0.0	878.36
9	1081.21	0.0	0.94	1099.88	0.0	1.45	1156.50	45.6	1106.66	0.0	934.47
10	1167.38	0.0	1.83	1150.42	0.0	4.84	1232.10	33.6	1159.78	0.0	1039.07
11	1274.67	0.0	1.94	1288.49	0.0	3.94	1422.41	38.0	1288.74	0.0	1094.11
12	1438.11	0.0	1.69	1443.00	0.0	3.77	1578.79	34.0	1453.82	0.0	1155.13
13	1485.67	0.0	2.72	1482.02	0.0	7.91	1624.16	35.3	1481.40	0.0	1287.18
14	1611.99	0.0	2.67	1658.55	0.0	7.42	1760.51	37.1	1624.96	0.0	1353.08
15	1748.31	0.0	2.66	1892.89	0.0	7.42	2105.02	33.6	1858.87	0.0	1457.61
16	1055.23	0.0	1.98	1383.57	0.0	3.53	1288.48	10.3	1267.87	0.0	1002.49
17	1117.22	0.0	1.64	1416.14	0.0	3.31	1314.09	9.4	1261.17	0.0	1042.35
18	1216.24	0.0	1.77	1614.11	0.0	2.91	1383.19	10.8	1366.21	0.0	1129.16
19	874.04	0.0	0.86	919.59	0.0	1.34	1146.74	22.0	969.96	0.0	813.50
20	950.72	0.0	0.78	972.76	0.0	1.24	1144.96	24.0	1140.47	0.0	848.93
21	1009.38	0.0	0.75	1096.08	0.0	1.31	1263.70	57.0	1174.43	0.0	909.06
Avg	1105.09	0.0	1.26	1171.38	5.24	2.68	1226.90	25.93	1166.86	0.0	959.14
ARPD	15.22	—	—	22.13	—	—	27.92	—	21.66	—	0.00

[a]Best solutions from 46 runs.
[b]Best solutions from n runs.
[c]Average solutions from 10 runs.
[d]Total times in seconds on a Pentium IV 1.5 GHz PC.
[e]Runtimes not available.

Figura 2: Comparación de heurísticas

Los resultados indican que la heurística **T-Cluster** domina a las otras heurísticas en la calidad de la solución mientras que los resultados de **T-Sweep** fueron aun ligeramente peores en comparación con la heurística **T-Cluster**. Pero aún así mejores que las de Chao. Fracasó en el diseño de soluciones viables para todos los problemas de prueba y tiene el inconveniente de que el número de inicios múltiples y, por lo tanto, el tiempo de ejecución aumenta con el número de clientes.

En 2011 Villegas [8] utilizó una metaheurística híbrida basada en un procedimiento de búsqueda adaptativo aleatorio codicioso (**GRASP**), una búsqueda de vecindario variable (**VNS**) y una reconexión de caminos (**PR**).

Comparison of GRASP/VNS with EvPR against other approaches from the literature.

Problem				Chao [7] (tabu search)		Scheuerer [45] (tabu search)				Lin et al. [32] (simulated annealing)				Caramia and Guerriero [5] (math. prog. heuristic)		GRASP/VNS + EvPR			
Number	n	BKS	Reference	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Cost	Gap	Best cost	Gap	Avg. Cost	Gap
1	50	564.68	[45]	565.02	0.06	566.80	0.38	567.98	0.59	566.82	0.38	568.86	0.74	566.80	0.38	**564.68**	0.00	565.99	0.23
2	50	611.53	[32]	662.84	8.39	615.66	0.68	619.35	1.28	612.75	0.20	617.48	0.97	620.15	1.41	**611.53**	0.00	614.23	0.44
3	50	618.04	[45]	664.73	7.56	620.78	0.44	629.59	1.87	**618.04**	0.00	620.50	0.40	632.48	2.34	**618.04**	0.00	618.04	0.00
4	75	798.53	[45]	857.84	7.43	801.60	0.38	809.13	1.33	808.84	1.29	817.71	2.40	803.32	0.60	**798.53**	0.00	803.51	0.62
5	75	839.62	[45]	949.98	13.14	**839.62**	0.00	858.98	2.31	**839.62**	0.00	858.95	2.30	842.50	0.34	**839.62**	0.00	841.63	0.24
6	75	930.64	[32]	1084.82	16.57	936.01	0.58	949.89	2.07	934.11	0.37	942.60	1.29	938.18	0.81	940.59	1.07	961.47	3.31
7	100	830.48	[45]	837.80	0.88	**830.48**	0.00	832.91	0.29	**830.48**	0.00	838.50	0.97	832.56	0.25	**830.48**	0.00	830.48	0.00
8	100	872.56	[32]	906.16	3.85	878.87	0.72	881.26	1.00	875.76	0.37	882.70	1.16	878.87	0.72	**872.56**	0.00	876.21	0.42
9	100	912.02	[32]	1000.27	9.68	942.31	3.32	955.95	4.82	912.64	0.07	921.97	1.09	980.42	7.50	914.23	0.24	918.45	0.71
10	150	1039.07	[45]	1076.88	3.64	1039.23	0.02	1052.65	1.31	1053.90	1.43	1074.38	3.40	1060.41	2.05	1046.71	0.74	1050.11	1.06
11	150	1093.37	(a)	1170.17	7.02	1098.84	0.50	1107.47	1.29	1093.57	0.02	1108.88	1.42	1170.70	7.07	**1093.37**	0.00	1100.95	0.69
12	150	1152.32	(a)	1217.01	5.61	1175.23	1.99	1184.58	2.80	1155.44	0.27	1166.59	1.24	1178.34	2.26	**1152.32**	0.00	1158.88	0.57
13	199	1287.18	[45]	1364.50	6.01	1288.46	0.10	1296.33	0.71	1320.21	2.57	1340.98	4.18	1288.46	0.10	1298.89	0.91	1305.83	1.45
14	199	1339.36	(a)	1464.20	9.32	1371.42	2.39	1384.13	3.34	1351.54	0.91	1367.91	2.13	1372.52	2.48	**1339.36**	0.00	1354.04	1.10
15	199	1420.72	(b)	1544.21	8.69	1459.55	2.73	1488.71	4.79	1436.78	1.13	1454.91	2.41	1470.21	3.48	1423.91	0.22	1437.52	1.18
16	120	1002.49	[45]	1064.89	6.22	**1002.49**	0.00	1003.00	0.05	1004.47	0.20	1007.26	0.48	1004.69	0.22	**1002.49**	0.00	1003.07	0.06
17	120	1026.20	[32]	1104.67	7.65	1042.35	1.57	1042.79	1.62	1026.88	0.07	1035.23	0.88	1042.35	1.57	1042.46	1.58	1042.61	1.60
18	120	1098.15	[32]	1202.00	9.46	1129.16	2.82	1141.94	3.99	1099.09	0.09	1110.13	1.09	1129.16	2.82	1113.07	1.36	1118.63	1.86
19	100	813.30	[32]	887.22	9.09	813.50	0.02	813.98	0.08	814.07	0.09	823.01	1.19	813.50	0.02	813.50	0.02	819.81	0.80
20	100	848.93	[45]	963.06	13.44	**848.93**	0.00	852.89	0.47	855.14	0.73	859.06	1.19	**848.93**	0.00	860.12	1.32	860.12	1.32
21	100	909.06	[45]	952.29	4.76	**909.06**	0.00	914.04	0.55	**909.06**	0.00	915.38	0.70	**909.06**	0.00	**909.06**	0.00	909.06	0.00
Average cost				1025.74				970.84				968.24		970.65				961.46	
Avg. gap above BKS					7.55%		0.89%		1.74%		0.48%		1.51%		1.74%		0.36%		0.84%
NBKS						5				4				2		12			

(a) GRASP/VNS + EvPR, (b) GRASP/VNS with PR (post-optimization).

Figura 3: Comparación de GRASP/VNS con EvPR frente a otros enfoques de la literatura

Como se puede ver en la figura (3) **GRASP/VNS** con **EvPR** supera al método de Scheuerer[12], logrando una pequeña brecha promedio a BKS de 0.84 y obteniendo 12 de 21 BKS con un solo conjunto de parámetros.Los valores en negrita en la tabla indican que el BKS se encontró mediante un método determinado.

La reconexión de rutas (**PR**) se introdujo en el contexto de la búsqueda tabú (**TS**) como un mecanismo para combinar la intensificación y la diversificación [23] . **PR** genera nuevas soluciones mediante la exploración de trayectorias que conectan soluciones de élite producidas previamente durante la búsqueda. La hibridación de **PR** con **GRASP** mejora el rendimiento de este último al abordar las críticas de falta de memoria que enfrenta el esquema básico de **GRASP**.
Luego en 2012 Derigs, Pullmann y Vogel [13] aplicaron una heurística que usa un enfoque de dos fases que comienza con la construcción de una solución factible inicial utilizando una estrategia específica del problema, seguida de una fase de mejora de la búsqueda de vecindarios en la que se utilizan varias búsquedas locales (**LS**) estándar y nuevas específicas del problema. así como los movimientos de búsqueda de vecindario grande (**LNS**) se aplican guiados por dos controles metaheurísticos simples como son las técnicas de recocido y la búsqueda tabú.
Derigs, Pullmann y Vogel[13] implementaron tres módulos denominados **LS-ABHC**, **LS-RRT** y **LNS-RRT**, que se concentran en un paradigma de vecindario único y que son la base de dos métodos híbridos que combinan la búsqueda local con la búsqueda de vecindarios grandes en el proceso de búsqueda. **HYBRID-ABHC** e **HYBRID-RRT** las cuales deciden aleatoriamente qué tipo de vecindario utilizar (**LS** o **LNS**) en cada iteración de la búsqueda

Finalmente en 2013 Villegas[9] utilizó una matemática bifásica simple, pero efectiva, que utiliza las rutas de los óptimos locales de un híbrido GRASP x̃ ILS como columnas en una formulación de partición de conjuntos del **TTRP**. Usando esta matemática Villegas resolvió tanto el **TTRP** clásico con flota fija como la nueva variante con flota ilimitada.
Villegas[9] dividió su mateheurística en dos fases. La fase GRASP x̃ ILS donde utiliza el enfoque introducido por Prins [11] , donde la fase de mejora se reemplaza por una búsqueda local iterada (ILS) y la Fase de gestión de grupos y partición de conjuntos donde las rutas se convierten en columnas en un problema de particiones

Instance				Simulated annealing				EvPR				Hybrid ABHC				Matheuristic (large pool)				Matheuristic (small pool)			
Id.	n	BKS	Ref.	Best	Avg.	Gap (%)	Time (min)	Best	Avg.	Gap (%)	Time (min)	Best	Avg.	Gap (%)	Time (min)	Best	Avg.	Gap (%)	Time (min)	Best	Avg.	Gap (%)	Time (min)
1	50	564.68	–[a]	566.82	568.86	0.74	9.51	**564.68**	565.99	0.23	1.17	**564.68**	564.81	0.02	1.73	**564.68**	564.82	0.03	1.32	**564.68**	565.53	0.15	0.32
2	50	611.53	–[b]	612.75	617.48	0.97	9.60	**611.53**	614.23	0.44	1.29	**611.53**	612.44	0.15	1.91	**611.53**	612.38	0.14	1.34	**611.53**	612.69	0.19	0.40
3	50	618.04	–[a]	**618.04**	620.50	0.40	11.24	**618.04**	618.04	0.00	1.05	**618.04**	618.04	0.00	1.91	**618.04**	618.04	0.00	1.09	**618.04**	618.04	0.00	0.57
4	75	798.53	–[a]	808.84	817.71	2.40	18.49	**798.53**	803.51	0.62	2.69	799.34	804.49	0.75	3.33	**798.53**	798.53	0.00	2.93	**798.53**	799.20	0.08	0.69
5	75	839.62	–[a]	**839.62**	858.95	2.30	15.16	**839.62**	841.63	0.24	2.82	**839.62**	839.62	0.00	3.64	**839.62**	839.62	0.00	2.60	**839.62**	839.62	0.00	0.88
6	75	930.64	–[b]	934.11	942.60	1.29	18.62	940.59	961.47	3.31	2.89	940.69	945.36	1.58	3.74	942.26	947.95	1.86	2.81	946.10	951.96	2.29	1.13
7	100	830.48	–[a]	**830.48**	838.50	0.97	33.60	**830.48**	830.48	0.00	6.05	**830.48**	830.48	0.00	7.67	**830.48**	830.55	0.01	13.77	**830.48**	830.82	0.04	3.04
8	100	870.94	–[c]	875.76	882.70	1.35	25.66	872.56	876.21	0.60	6.96	871.98	876.45	0.63	8.18	**870.94**	872.36	0.16	7.00	**870.94**	875.01	0.47	2.15
9	100	912.02	–[b]	912.64	921.97	1.09	30.47	914.23	918.45	0.71	8.38	922.36	926.14	1.55	7.89	914.23	914.23	0.24	8.03	914.23	914.78	0.30	2.08
10	150	1036.20	–[e]	1053.90	1074.38	3.69	60.94	1046.71	1050.11	1.34	18.84	1040.46	1046.58	1.00	14.19	**1036.20**	1037.83	0.16	30.79	1036.96	1044.04	0.76	17.87
11	150	1091.91	–[c]	1093.57	1108.88	1.55	56.17	1093.37	1100.95	0.83	21.20	1093.89	1102.07	0.93	17.38	1092.22	1093.29	0.13	25.20	1094.16	1098.68	0.62	6.31
12	150	1149.41	–[c]	1155.44	1166.59	1.49	63.71	1152.32	1158.88	0.82	25.78	1154.82	1170.32	1.82	18.04	1149.75	1149.95	0.05	28.77	1150.41	1152.00	0.23	6.91
13	199	1284.71	–[c]	1320.21	1340.98	4.38	165.41	1298.89	1305.83	1.64	43.94	1289.20	1319.07	2.67	28.03	1285.78	1287.41	0.21	86.15	1288.37	1293.55	0.69	17.14
14	199	1333.66	–[c]	1351.54	1367.91	2.57	132.06	1339.36	1354.04	1.53	45.57	1341.25	1352.50	1.41	28.47	**1333.66**	1336.62	0.22	28.49	1339.83	1342.18	0.64	10.58
15	199	1416.51	–[c]	1436.78	1454.91	2.71	154.10	1423.91	1437.52	1.48	59.83	1423.55	1452.25	2.52	30.58	**1416.51**	1418.52	0.14	36.91	1417.88	1422.74	0.44	13.24
16	120	1000.84	–[e]	1004.47	1007.26	0.64	43.14	1002.41	1002.80	0.20	15.07	1001.48	1026.59	2.57	9.71	**1000.84**	1002.19	0.13	10.76	1002.22	1003.27	0.24	4.60
17	120	1026.17	–[d]	1026.88	1035.23	0.88	33.73	**1026.17**	1026.65	0.05	12.62	1026.20	1031.81	0.55	10.49	**1026.17**	1026.33	0.02	10.61	1026.35	1026.35	0.02	4.08
18	120	1098.15	–[d]	1099.09	1110.13	1.09	31.78	1098.55	1102.43	0.39	12.62	**1098.15**	1098.46	0.03	11.13	1100.84	1106.51	0.76	10.70	1103.09	1109.49	1.03	4.17
19	100	812.69	–[d]	814.07	823.01	1.27	28.84	**812.69**	820.21	0.93	4.94	**812.69**	819.43	0.83	6.96	**812.69**	814.05	0.17	4.86	812.91	814.38	0.21	3.11
20	100	848.12	–[d]	855.14	859.06	1.29	24.57	859.31	859.31	1.32	5.48	**848.12**	848.20	0.01	8.22	**848.12**	849.31	0.14	5.39	**848.12**	849.66	0.18	3.28
21	100	909.06	–[a]	**909.06**	915.38	0.70	26.84	**909.06**	909.06	0.00	6.25	**909.06**	909.25	0.02	9.45	**909.06**	909.06	0.00	6.02	**909.06**	909.06	0.00	3.07
Number of BKS				4				9				9				15				9			
Avg. gap (%) or Avg. time (min)						1.61	47.32			0.79	14.55			0.91	11.08			0.22	15.50			0.41	5.03
Max. gap (%) or time (min)						4.38	165.41			3.31	59.83			2.67	30.58			1.86	86.15			2.29	17.87

[a] Scheuerer et al. [56].
[b] Lin et al. [38].
[c] Villegas et al. [67].
[d] Derigs et al. [16].
[e] BKS improved.

Figura 4: Resultados detallados de la pruebas realizadas para el TTRP

Como se observa en la figura (4) esta matemática supera a los métodos más avanzados tanto en términos de calidad de la solución como de tiempo de cómputo. Los resultados subrayados indican una composición diferente de la flota con respecto a las mejores soluciones encontradas por Simulated annealing.

4. Modelo Matemático

4.1. Modelo de Chao[2]

En primer lugar, se selecciona un punto inicial para cada m_l ruta de vehículo pura o completa y m_k - m_l ruta de camión pura
Las variables son:

- d_{ij} Costo de asignar al cliente i a la ruta del vehículo o camión i
- x_{ij} Parámetro binario que toma el valor 1 si se asigna al cliente i a la ruta vehículo o camión j, toma el valor de 0, en caso contrario

La función objetivo minimiza el costo total de asignación de la siguiente manera:

$$\text{mín} \sum_{i=1}^{n} \sum_{j=1}^{m_k} d_{ij} x_{ij} \tag{1}$$

sujeto a :

$$\text{mín} \sum_{j=1}^{m_k} x_{ij} = 1, i = 1, 2, ..., n. \tag{2}$$

Las restricciones que debe cumplir son:

$$\sum_{i=1}^{n} q_i x_{ij} \leq Q_1 + Q_k, j = 1, 2, ..., n, \tag{3}$$

$$\sum_{i=1}^{n} q_i x_{ij} \leq Q_k, j = m_1 + 1, ..., m_k, \tag{4}$$

$$x_{ij} = \{0, 1\}\, i = 1, 2, ..., n, j = 1, 2, ..., m_k, \tag{5}$$

$$0 \leq x_{ij}, i = 1, 2, ..., n, j = 1, 2, ..., m_k, \tag{6}$$

$$n - n_1 \leq m_k, \tag{7}$$

$$n + m_k - n_1 \leq 2m_k. \tag{8}$$

- (3) Obliga a que la demanda máxima de carga de cada ruta vehicular pura o completa sea menor o igual a la suma de las capacidades de un camión y un remolque
- (4) Obliga a que la carga de demanda máxima de cada ruta de camión sea menor o igual a la capacidad del camión.
- (5) Restricción de integralidad
- (6) Relaja (4) obligando a los x_{ij} a tener valores dentro del intervalo cerrado entre 0 y 1.
- (7) Es un límite en el número de variables x_{ij} de valor fraccionario
- (8) Es el número de clientes que tienen variables x_{ij} de valor fraccionario

4.2. Modelo de Villegas[9]

Conjuntos:

- Sea Ω el conjunto de rutas factibles en un **TTRP**
- Sea $\Gamma \subseteq \Omega$ el conjunto de rutas puras de camiones realizadas por camiones que visitan clientes en N_t y N_ν
- Sea $\Psi \subseteq \Omega$ el conjunto de rutas de vehículos puros realizadas por vehículos completos que sirven solo a clientes en N_ν
- Sea $\mathit{\Pi} \subseteq \Omega$ el conjunto de rutas vehiculares con viajes de segundo nivel(sub-recorrido)

Las variables y parámetros son:

- a_{ij} parámetro binario que toma el valor de 1 si el cliente i $\in$ N es visitado en una ruta de camiones pura j $\in \Gamma$ toma el valor de 0, en caso contrario
- b_{ik} parámetro binario que toma el valor de 1 si el cliente i $\in N_\nu$ es visitado en una ruta de vehículo pura k $\in \Psi$ toma el valor de 0, en caso contrario
- e_{ik} parámetro binario que toma el valor de 1 si el cliente i $\in N$ es visitado en en ruta vehicular con viajes de segundo nivel l $\in \mathit{\Pi}$ toma el valor de 0, en caso contrario
- x_j variable binaria que toma el valor de 1 si y solo si se utiliza la ruta j $\in \Gamma$ en la solución del **TTRP**
- y_k variable binaria que toma el valor de 1 si y solo si se utiliza la ruta k $\in \Psi$ en la solución del **TTRP**

- z_l variable binaria que toma el valor de 1 si y solo la ruta l $\in \Pi$ está incluida en la solución del **TTRP**
- c_r variable que representa el costo (distancia total) de cualquier ruta $r \in \Omega(= \Gamma \cup \Psi \cup \Pi)$

La función objetivo minimiza la distancia total de la solución de la siguiente manera:

$$\text{mín}\, z = \sum_{j \in \Gamma} c_j x_j + \sum_{k \in \Psi} c_k y_k + \sum_{l \in \Pi} c_l z_l \tag{1}$$

sujeta a:

$$\text{mín}\, z = \sum_{j \in \Gamma} a_{ij} x_j + \sum_{k \in \Psi} b_{ik} y_k + \sum_{l \in \Pi} e_{il} z_l \tag{2}$$

Las restricciones que debe cumplir son:

$$\sum_{j \in \Gamma} a_{ij} x_j + \sum_{l \in \Pi} e_{il} z_l = 1 \forall i \in N_\nu \tag{3}$$

$$\sum_{j \in \Gamma} x_i + \sum_{k \in \Psi} y_k + \sum_{l \in \Pi} z_l \leq m_t \tag{4}$$

$$\sum_{k \in \Psi} y_k + \sum_{l \in \Pi} z_l \leq m_r \tag{5}$$

$$x_j \in \{0, 1\}, \forall j \in \Gamma \tag{6}$$

$$y_k \in \{0, 1\}, \forall k \in \Psi \tag{7}$$

$$z_l \in \{0, 1\}, \forall l \in \Pi \tag{8}$$

- (2) Asegura que cada cliente de vehículos sea visitado exactamente una vez.
- (3) Asegura que cada cliente de camiones sea visitado exactamente una vez por una ruta de camiones o en un viaje de segundo nivel de una ruta de vehículos con este tipo de viajes.
- (4) Impone un límite superior al número de camiones
- (5) Impone un límite superior al número de remolques
- (6) Dominio de x
- (7) Dominio de y
- (8) Dominio de z

Dado que la cantidad de rutas factibles es enorme, usar el modelo directamente a menudo no es práctico. Además, la estructura de rutas vehiculares con viajes de segundo nivel implica un subproblema de tarificación muy complejo cuando las **TTRP** se resuelven utilizando métodos tradicionales de generación de columnas o sucursales y precios

5. Representación

Las estructuras de datos utilizadas son:

- Clientes: Estructura que almacena lo posición en el plano del cliente, su id, si es que fue visitado o no y su tipo de cliente
- Tráiler: Estructura que almacena la id del cliente que esta visitando el tráiler, la capacidad total del tráiler, la carga que transporta, el tour que realizo y las funciones de añadir cliente y comprobar carga.
- Camión: Estructura que almacena id del cliente que esta atendiendo el camión, su capacidad total y su carga actual, el tour realizado además de las funciones de comprobar carga y añadir cliente
- Grafo: Estructura que almacena la cantidad de camiones, cantidad de clientes y cantidad de tráileres, la matriz de distancias y todas las funciones necesarias para los distintos algoritmos utilizados

Una solución está representada por una cadena de números que consta de una permutación de n clientes denotada por el conjunto de $1, 2, ..., n$ y N ceros ficticios (depósitos) seguidos por los tipos de vehículos de servicio de los VC individuales. los N ceros ficticios se usan para separar rutas o terminar un sub-tour.

El primer número de la solución indica el primer cliente al que se atenderá en la primera ruta. Si el primer cliente de una ruta va a ser atendido por un solo camión, la ruta se configura como un **PTR**. Otros clientes se agregan a la ruta uno por uno de izquierda a derecha para representar la secuencia en la que son atendidos, siempre que no se viole la capacidad del vehículo en uso.

visualmente la representacion se ve de la siguiente forma:

0	4	0	11	8	1	0	15	0	0	9	10	2	3	0
Ruta 1									Ruta 2					

Donde la i-esima celda corresponde al arreglo de clientes de la ruta + en el caso de la ruta 1 los clientes 11, 8 y 1 pertenecen a la ruta de camión mientras que los clientes 4 y 15 corresponden a los clientes de vehículo
Como podemos observar las rutas estan separadas por estos ceros "ficticios"mencionados anteriormente. por eso mediante el uso de las estructuras mencionadas y ciertos algoritmos nuestro programa entrega un ouput de este estilo:

Trailer	0	4	15	0	
Camion	0	11	8	1	0

6. Descripción del algoritmo

La solución inicial se genera a través del algoritmo greedy, el cual mientras existan clientes que no hayan sido visitados, este seguirá ejecutándose, en el algoritmo priorizamos la asignación de los tráileres sobre los camiones ya que estos tienen la restricción de que existen ciertos lugares que no pueden visitar. Tomaremos al cliente mas cercano a la posición actual y averiguaremos que tipo de cliente es, ya sea cliente de vehículo o de camión. en base a esta información sabremos si el tráiler sigue con el camión o tenemos que desacoplarlo. Finalmente retornamos la ruta realizada

```
def greedy (clientes)
 while clientes != todos visitados
    for cliente in clientes
        if cliente no fue visitado
            if quedan trailers por asignar
                elegir el cliente mas cercano a nuestra posicion actual
                    if client.tipo_cliente = 0
                        if carga_trailer + demanda_cliente <= capacidad trailer
                            anadir cliente a ruta de trailer
                        else
                            if el trailer esta lleno
                                devolver trailer al deposito
                    else
                        if carga_actual_camion + demanda_cliente <= capacidad camion
                            trailer se queda en el cliente anterior
                            anadir cliente a ruta de camion
                        else
                            devolver camion al trailer
            else
                elegir al cliente mas cercano a la posicion actual del camion
                y anadirlo a ruta de camion
    return rutas
```

La tecnica reparadora a utilizar con la solucion inicial es el algoritmo simulated anneling donde **X** es la solución resultante de aplicar el algoritmo greedy, **temperatura_final** es la mínima temperatura que debe existir para que el programa siga iterando,**iter** es la cantidad de iteraciones que deben pasar para que la temperatura disminuya.

En cada iteración, la siguiente solución **Y** se genera a partir de **X** y se evalúa su valor de función objetivo. En cada iteración, la probabilidad de reemplazar **X** por **Y** , donde **X** es la solución actual e **Y** es la siguiente solución cuando **Y** es peor que **X** tomamos en valor **r** el cual es un numero aleatorio entre 0 y 1 y lo comparamos con exp((obj(Y,P) - obj(X,P)) / T) en el caso que r sea menor a esta exponencial reemplazaremos **X** por **Y** para así explorar un poco mas el espacio de búsqueda.

La temperatura disminuye luego de **iter** iteraciones desde la disminución anterior

El movimiento utilizado es el swap aleatorio, ya que tenemos un arreglo de arreglos donde la i-esima celda son los clientes de la ruta i + 1, nuestro swap toma dos clientes aleatoriamente elegidos de dos celtas aleatoriamente elegidas y los intercambia(solo intercambia clientes de tráiler con clientes de tráiler y clientes de camión con clientes de camión).

```
def SA(X, iteraciones , T_inicial)
```

```
iter igual a numero_aleatorio
temperatura_final igual a 0
temperatura_descuento igual a numero random
valor_solucion igual a obj(X, Penalizacion)
mejor_x igual a X
while Temperatura_inicial disntinto de  temperatura_final
    iteraciones igual a iteraciones + 1
    r igual a random(0,1);
    if r < 0.5
        Y es igual a Nueva solucion generada con el movimiento swap aleatorio sobre
        X.camiones
    else
        Y es igual a Nueva solucion generada con el movimiento swap aleatorio sobre
        X.traileres
    if obj(Y,P) - obj(X,P) > 0
            X = Y
    else
        if r < exp(obj(Y,P) - obj(X,P)/KT)
            X igual Y
    if (obj(X,P) < valor_solucion and X es factible)
        mejor_x igual a  X
        valor_solucion igual a obj(X,P)
    if iteraciones es igual a iter
        T igual a T - temperatura_descuento
```

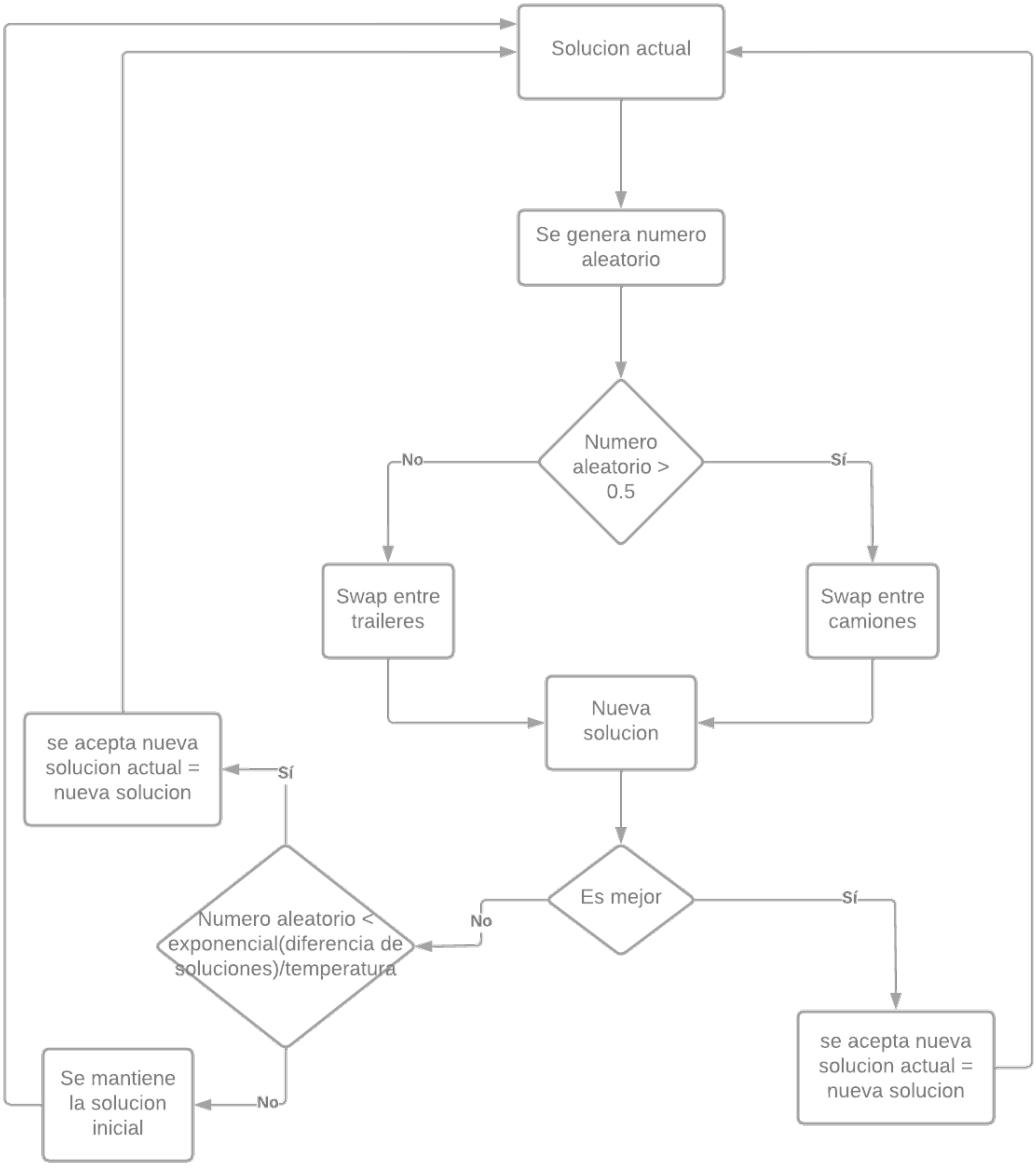

Figura 5: Diagrama de flujo del algoritmo sa utilizado

7. Experimentos

Los experimentos se realizaron en un pc que cuenta con windows 11 de sistema operativo y que posee un Ryzen 5 3600 junto con 24GB ram el cual denominaremos **Entorno 1** y en un notebook 80UD Lenovo ideapad 110-15ISK el cual cuenta con Manjaro Linux x86 de sistema operativo y posee un procesador Intel i3-6006U (4) @ 2.000GHz con 2 GB de ram el cual denominaremos **Entorno 2**

los parámetros utilizados son temperatura máxima utilizada y iteraciones máximas para que se detenga el algoritmo. Estos dos los obtuve de forma totalmente aleatoria en mis experimentos.

3 de los 4 experimentos expuestos se realizaron para observar que pasaba si movíamos los parámetros necesarios para el algoritmo sa. Mientras que el experimento numero **1** se realizo para observar como se comporta en algoritmo en diferentes estaciones de trabajo.

experimento	Temperatura	iteraciones	instancia
1	100	1000	diferentes instancias
2	100	diferentes	15
3	diferentes	1000	15
4	diferentes	diferentes	15

Cabe destacar que se realizaron mas experimentos como los mostrados, pero con diferentes instancias. los resultados fueron iguales y los mas significativos se dieron con la instancia 15

8. Resultados

Experimento numero 1

	Cantidad de iteraciones		Tiempo de ejecución[ms]	
instancias	Entorno 1	Entorno 2	Entorno 1	Entorno 2
01	1057	1114	2.1	10.9
02	1114	1114	3	10.8
08	1222	1222	4.9	56.2
15	1430	1438	9.4	119.1

Del experimento numero 1 podemos concluir que los componentes afectan de manera directa al tiempo de ejecución del algoritmo no así con las iteraciones totales de este(esto debido a que las iteraciones dependían mas de la aleatoriedad del sa). Podemos notar que la estación de trabajo 1 es mucho mas rápida al momento de ejecutar el algoritmo, en algunos casos incluso siendo 10 veces mas rápida.

Experimento numero 2

	Cantidad de iteraciones sa			
Cantidad de iteraciones sa	135	305	840100	1
solucion[km]	2487.5	2487.5	2487.5	2487.5
tiempo[ms]	4.7	5.3	4707	4.4
iteraciones totales	573	743	840538	439

Del experimento numero 2 podemos concluir que la cantidad de iteraciones que ejecuta el sa no afectan a la solución lograda. Esto sucede ya que nuestra solución inicial esta realizada con greedy el cual ya nos da un óptimo local o global en algunos casos por lo cual al momento de realizar la búsqueda de nuevas soluciones nuestro algoritmo queda estancado rápidamente.(notar que a veces las iteraciones totales son mayores a ingresadas para el algoritmo sa. Esto sucede porque se suman las iteraciones de sa junto a las de greedy)

experimento numero 3

	Temperatura			
Temperatura	445	807	285	783
solucion[km]	2487.5	2487.5	2487.5	2487.5
tiempo[ms]	9.5	9.1	9.5	9.3
iteraciones totales	1438	1438	1438	1438

Del experimento numero 3 podemos concluir que la temperatura con la que inicia el sa no afecta a la solución lograda. Esto sucede ya que nuestra solución inicial esta realizada con greedy el cual ya nos da un óptimo local o global en algunos casos por lo cual al momento de realizar la búsqueda de nuevas soluciones nuestro algoritmo queda estancado rápidamente.(notar que a veces las iteraciones totales son iguales ya que estas dependen de las iteraciones greedy y las ejecutadas por sa)

experimento numero 4

Temperatura	937	35	286	519
iteraciones sa	700	519	523	786
solucion lograda[km]	2487.5	2487.5	2487.5	2487.5
tiempo [ms]	10.4	8.7	8.6	11.1
iteraciones totales	1138	957	970	1224

Del experimento numero 4 podemos deducir que al cambiar las iteraciones máximas a ejecutar por sa y la temperatura inicial de este no variamos los resultados de solución encontrada. Esto sucede ya que nuestra solución inicial esta realizada con greedy el cual ya nos da un óptimo local o global en algunos casos por lo cual al momento de realizar la búsqueda de nuevas soluciones nuestro algoritmo queda estancado rápidamente.

Los demás experimentos ejecutados nos llevan a los mismos resultados que los ya expuestos.

9. Conclusiones

En este informe primeramente se explicó en que consiste el **TTRP**, además de analizar las diferentes técnicas que se han utilizado a lo largo de los años donde vemos que de primera todas las técnicas resuelven el **TTRP**, pero existen algunas que de igual forma sirven para resolver otros problemas:
por ejemplo la técnica de Chao[2] sirve de igual forma para resolver el problema **TSTTRP** mientras que la primera técnica de Villegas[8] sirve de igual forma para resolver el problema **STTRPSD** por otro lado la técnica de Derings[14] sirve para resolver el problema **TTRPTW** y la segunda técnica de Villegas[8] sirve para resolver el problema **RTTRP**
Si bien en un comienzo todas las técnicas eran parecidas ya que todas usaban heurísticas cada nueva técnica era una mejora de la anterior por lo que en la base eran ïguales"pero cada técnica implementaba una "novedad"que la hacía mejor que la anterior. Cuando Villegas plantea su

segunda técnica ocurre una revolución en la tendencia de cómo resolver el problema ya que no uso una heurística en su técnica si no una mate heurística por lo que se puede concluir que la segunda técnica de Villegas es la más prometedora ya que la nueva matemática planteada en su técnica demuestra ser muy efectiva y eficiente, reduciendo el tiempo y diferencia entre los BKS
En conclusion el **TTRP** es un problema muy real el cual es muy óptimo para ser resuelto con algún algoritmo de inteligencia artificial. Como ya vimos utilizar una solución obtenida con un algoritmo de construcción de soluciones como es greedy para una técnica reparadora como es sa no es una buena opción ya que nuestro algoritmo quedara estancado rápidamente.
¿como podríamos mejorar esto en el futuro?
Una opción viable es comparar soluciones obtenidas con un algoritmo de construcción de soluciones con la obtenida de algún algoritmo reparador con solución inicial aleatoria. otra solución podría ocupar otro movimiento al que yo utilice para realizar el sa. Como en todo proceso humano siempre hay algo que se puede mejorar y como los algoritmos de ia están en constante mejora el futuro para este tipo de problemas es prometedor. Solo hay que experimentar ya sea con los algoritmos ya obtenidos o analizar y buscarle mejoras a los últimos avances, como son las técnicas de Villegas.

Referencias

[1] Leland R. Miller Billy E. Gillett. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):205–451, 1974.

[2] I-Ming Chao. A tabu search method for the truck and trailer routing problem. *Computers Operations Research*, 29(1):33–951, 2002.

[3] Michael Drexl. Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem. *Revista de Metodos Cuantitativos para la Economia y la Empresa*, 12(1):5–38, 2011.

[4] F.Glover. Future paths for integer programming and links to artificial intelligence. *Computers Operations Research*, 41:3–38, 1996.

[5] J. H. Ramser G. B. Dantzig. The truck dispatching problem. *Management Science*, 6(1):1–140, 1959.

[6] A Hoff. Heuristics for rich vehicle routing problems. *Doctoral dissertation, Ph. D. Thesis. Molde University College*, 2006.

[7] johanna c. gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93(1):135–147, 1996.

[8] Caroline Prodhon Andrés L.Medaglia-Nubia Velasco Juan G.Villegas, Christian Prins. A grasp with evolutionary path relinking for the truck and trailer routing problem. *Computers Operations Research*, 38(9):1319–1334, 2011.

[9] Caroline Prodhon Andrés L.Medaglia-Nubia Velasco Juan G.Villegas, Christian Prins. A matheuristic for the truck and trailer routing problem. *Computers Operations Research*, 230(2):231–244, 2013.

[10] Hansen P. The steepest ascent mildest descent heuristic for combinatorial programming. *Talk presented at the Congress on numerical methods in combinatorial optimization*, 1986.

[11] Christian Prins. A grasp x̃ evolutionary local search hybrid for the vehicle routing problem. *Studies in Computational Intelligence*, 161:35–53, 2009.

[12] Stephan Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers Operations Research*, 33(4):894–909, 2006.

[13] Vincent F. Yu y Shuo-Yan Chou Shih-Wei Lin. A note on the truck and trailer routing problem. *Expert Systems with Applications*, 37(1):899–903, 2010.

[14] Markus Pullmann y Ulrich Vogel Ulrich Derigs. Truck and trailer routing problems, heuristics and computational experience. *Computers Operations Research*, 40(2):536–546, 2013.