PROBLEMAS DE ALTA COMPLEJIDAD

Dr. Flabio Gutierrez

flabiogs@yahoo.es

https://flabiogutierrez.wordpress.com/

a. Búsquedas por Internet

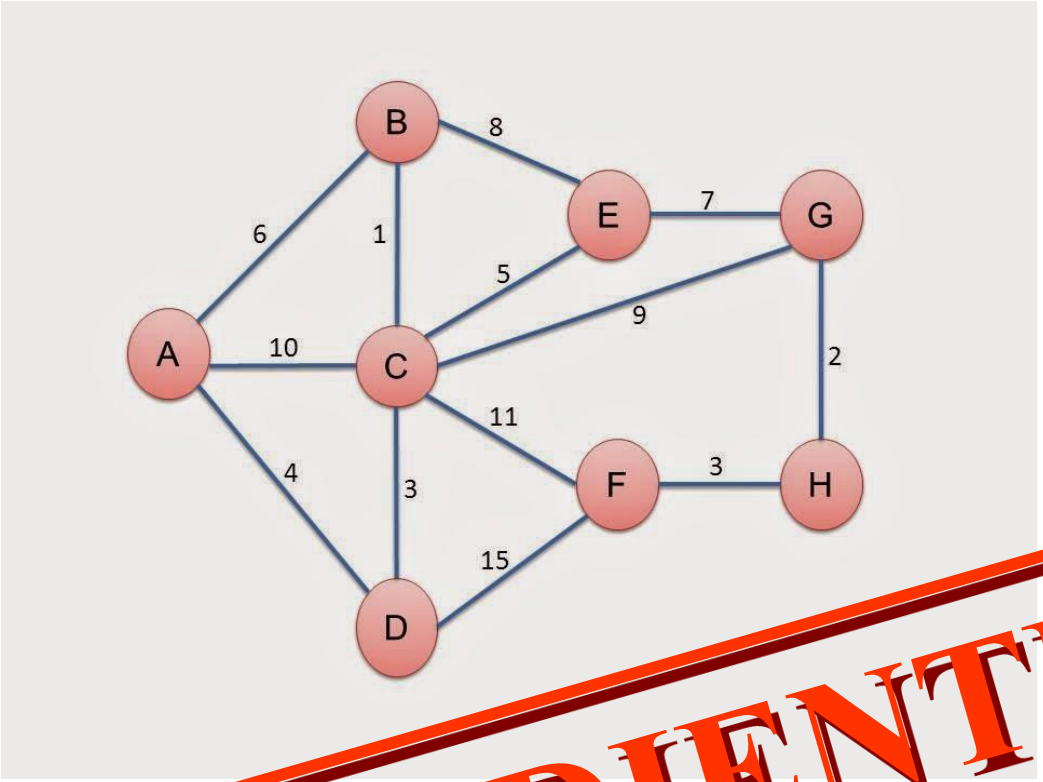
b) Planificador de rutas

Mito:

La capacidad de computación es ilimitada, ¿por qué preocuparse de la eficiencia si siempre podemos encontrar un sistema más potente?

Además está la Ley de Moore

1.- EL PROBLEMA DE AGENTE VIAJERO
TRAVELING SALESMAN PROBLEM (TSP)



PENDIENTE

1.- EL PROBLEMA DE AGENTE VIAJERO
TRAVELING SALESMAN PROBLEM (TSP)

Sea n= número de ciudades
Para n=7, hay 7! = 5040 posibles rutas (permutaciones) posibles.
Para n=10, hay 10! = 3 628 800 rutas posibles.
Para n=12, hay 12! = 479 001 600 rutas posibles.
Para n=20 hay 2432 9020 0817 6640 000 rutas posibles.

Explorando 10000 permutaciones por segundo, una búsqueda exhaustiva demandaría un tiempo estimado de:

n	Rutas	Tiempo estimado
7	5040	0.504
10	3 628 800	362.88 aprox 6.048 minutos
12	479 001 600	47 900.2 = 798.333 minutos 13.3055 horas
20	2432 9020 0817 6640 000	243290200817664 segundos 4.05484xE10 minutos 7.82183xE6 años

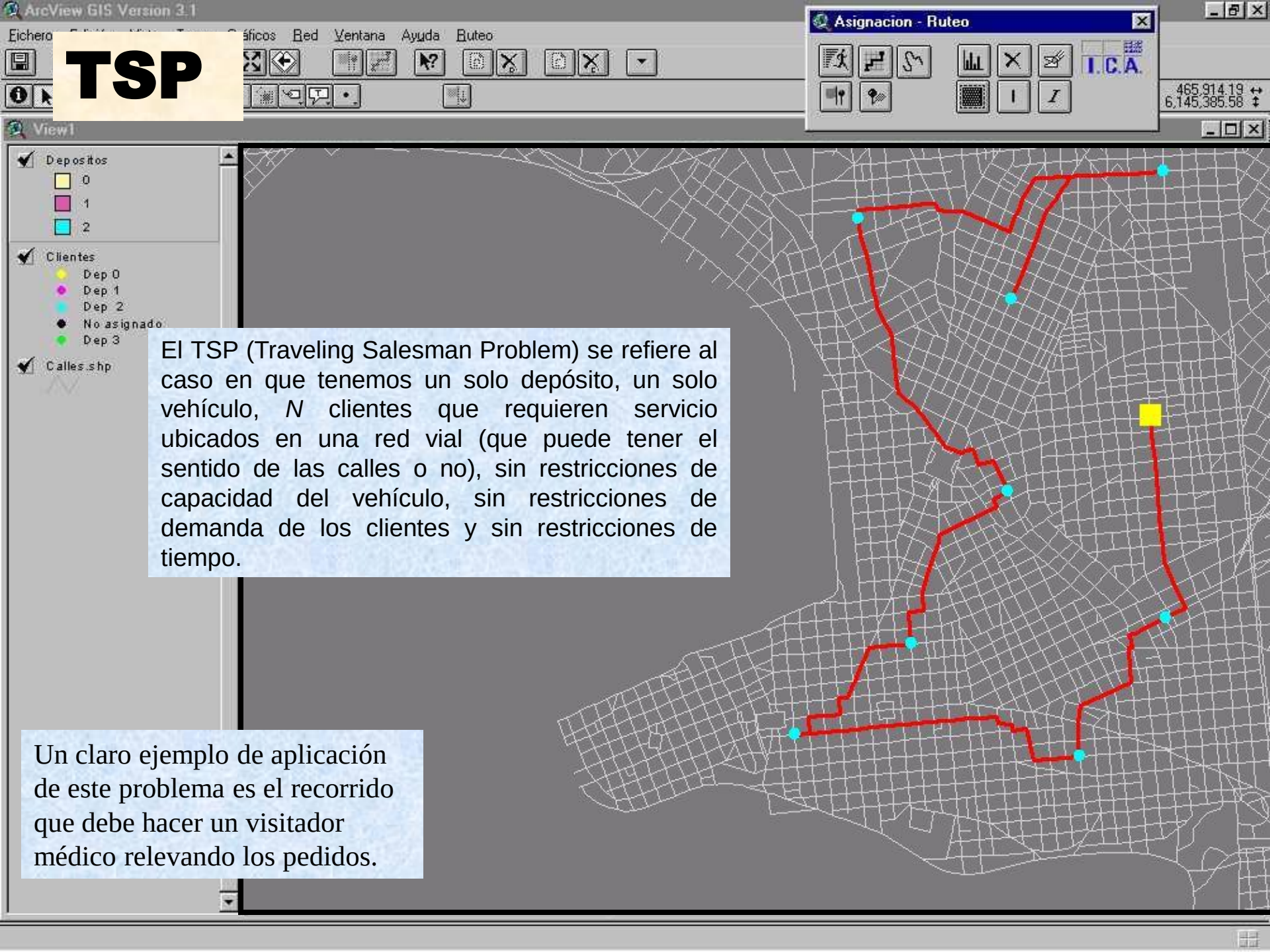
1.- EL PROBLEMA DE AGENTE VIAJERO
TRAVELING SALESMAN PROBLEM (TSP)

Aplicaciones

El camino más corto entre varios puntos,

- Un plan de mínimo coste para repartir mercancías a clientes,
- Una asignación óptima de trabajadores a tareas a realizar,
- Una secuencia óptima de proceso de trabajos en una cadena de producción,
- Una distribución de tripulaciones de aviones con mínimo coste,
- El mejor enrutamiento de un paquete de datos en Internet

TSP



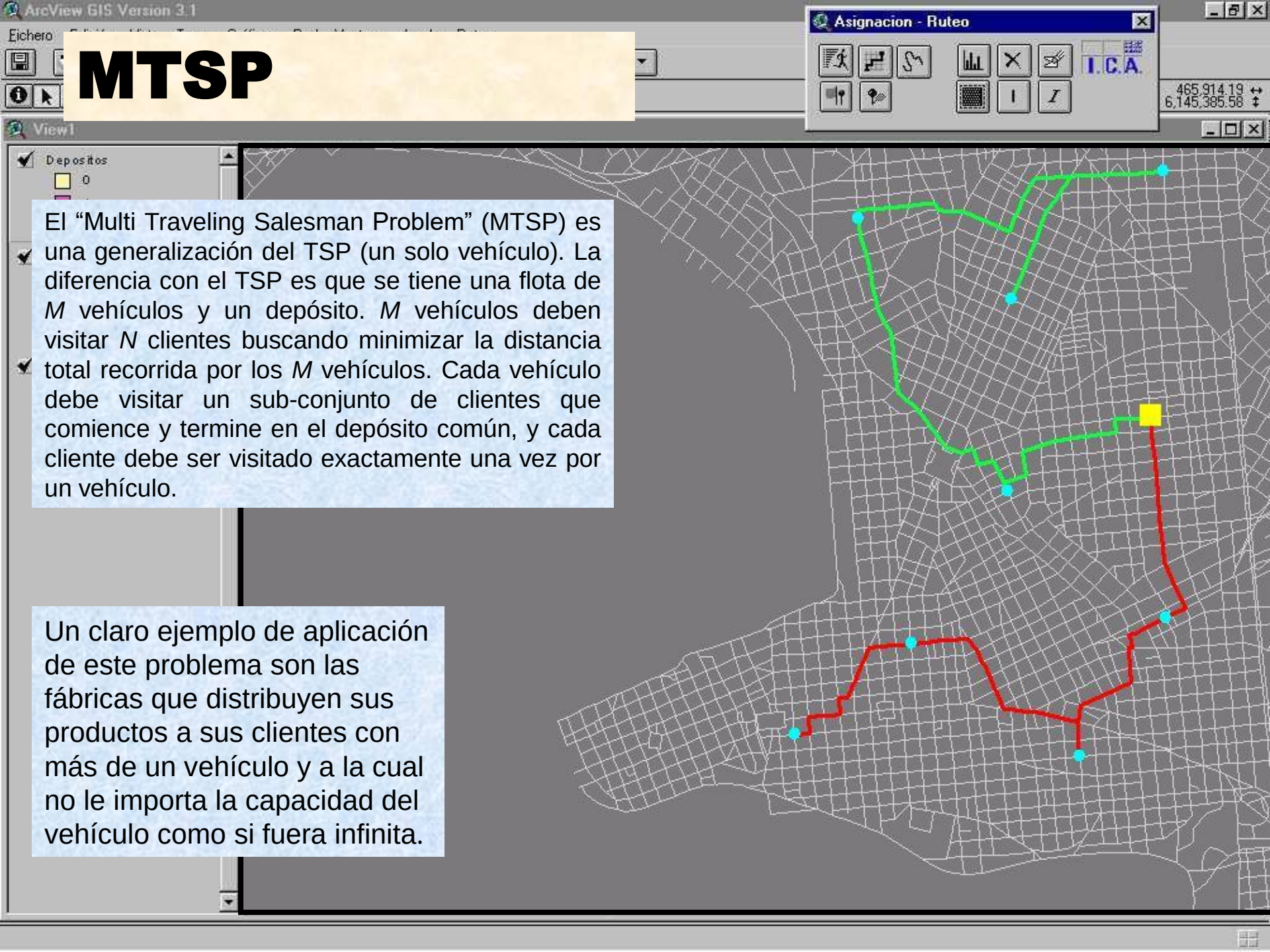
El TSP (Traveling Salesman Problem) se refiere al caso en que tenemos un solo depósito, un solo vehículo, *N* clientes que requieren servicio ubicados en una red vial (que puede tener el sentido de las calles o no), sin restricciones de capacidad del vehículo, sin restricciones de demanda de los clientes y sin restricciones de tiempo.

Un claro ejemplo de aplicación de este problema es el recorrido que debe hacer un visitador médico relevando los pedidos.

MTSP

El “Multi Traveling Salesman Problem” (MTSP) es una generalización del TSP (un solo vehículo). La diferencia con el TSP es que se tiene una flota de *M* vehículos y un depósito. *M* vehículos deben visitar *N* clientes buscando minimizar la distancia total recorrida por los *M* vehículos. Cada vehículo debe visitar un sub-conjunto de clientes que comience y termine en el depósito común, y cada cliente debe ser visitado exactamente una vez por un vehículo.

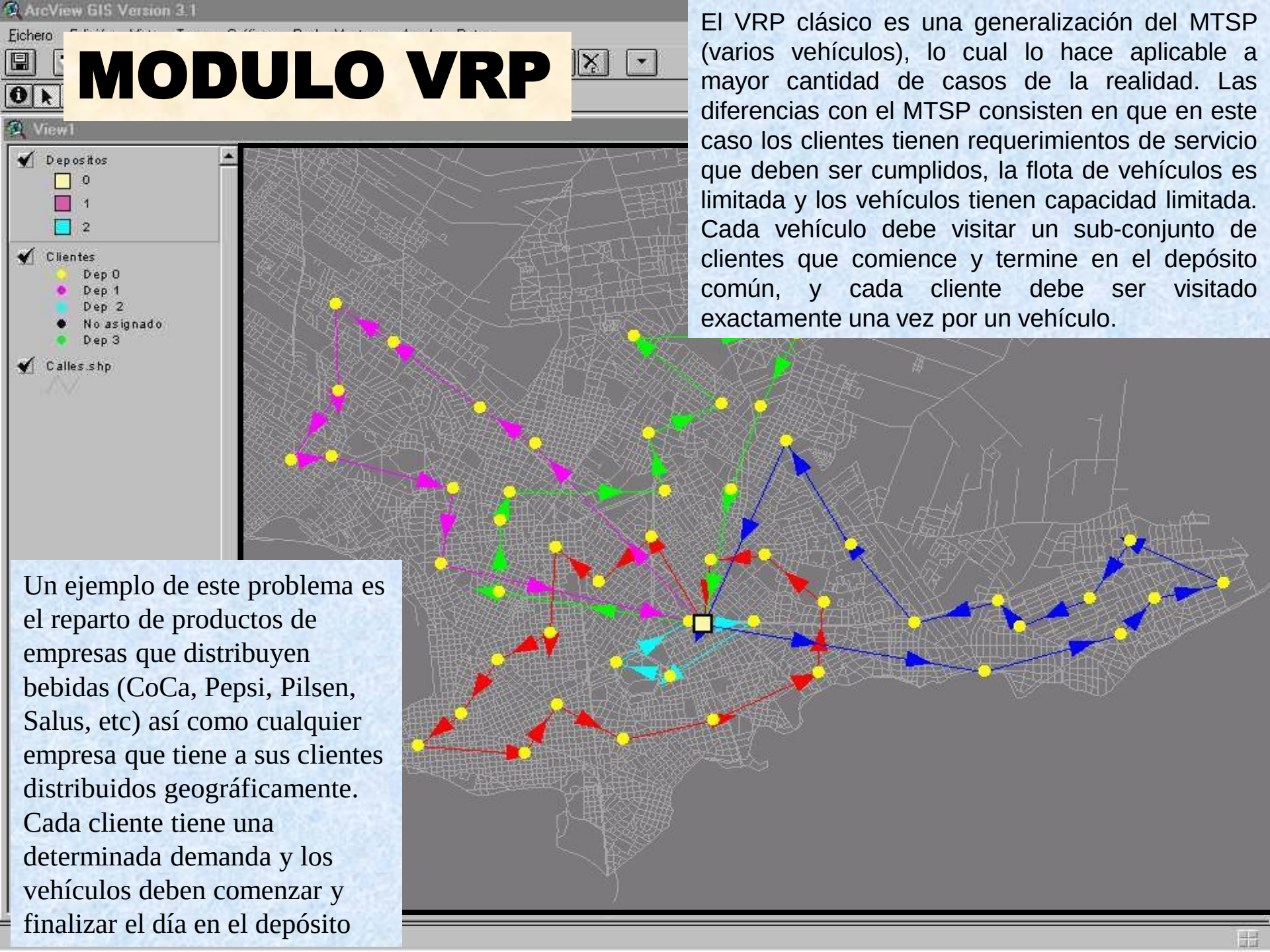
Un claro ejemplo de aplicación de este problema son las fábricas que distribuyen sus productos a sus clientes con más de un vehículo y a la cual no le importa la capacidad del vehículo como si fuera infinita.



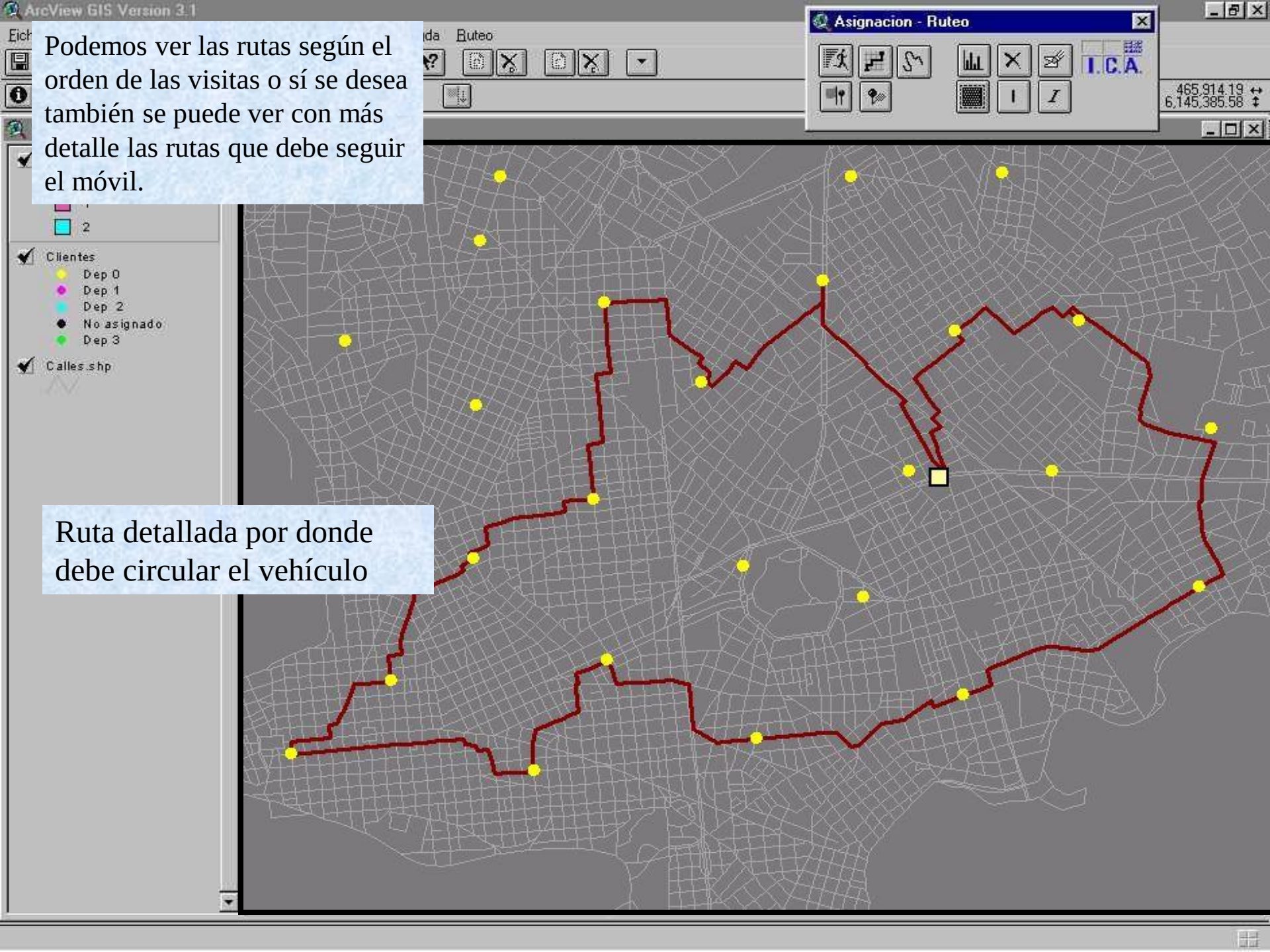
2.- EL PROBLEMA DEL RUTEO DE VEHÍCULOS
VEHICLE ROUTING PROBLEM (VRP)

MODULO VRP

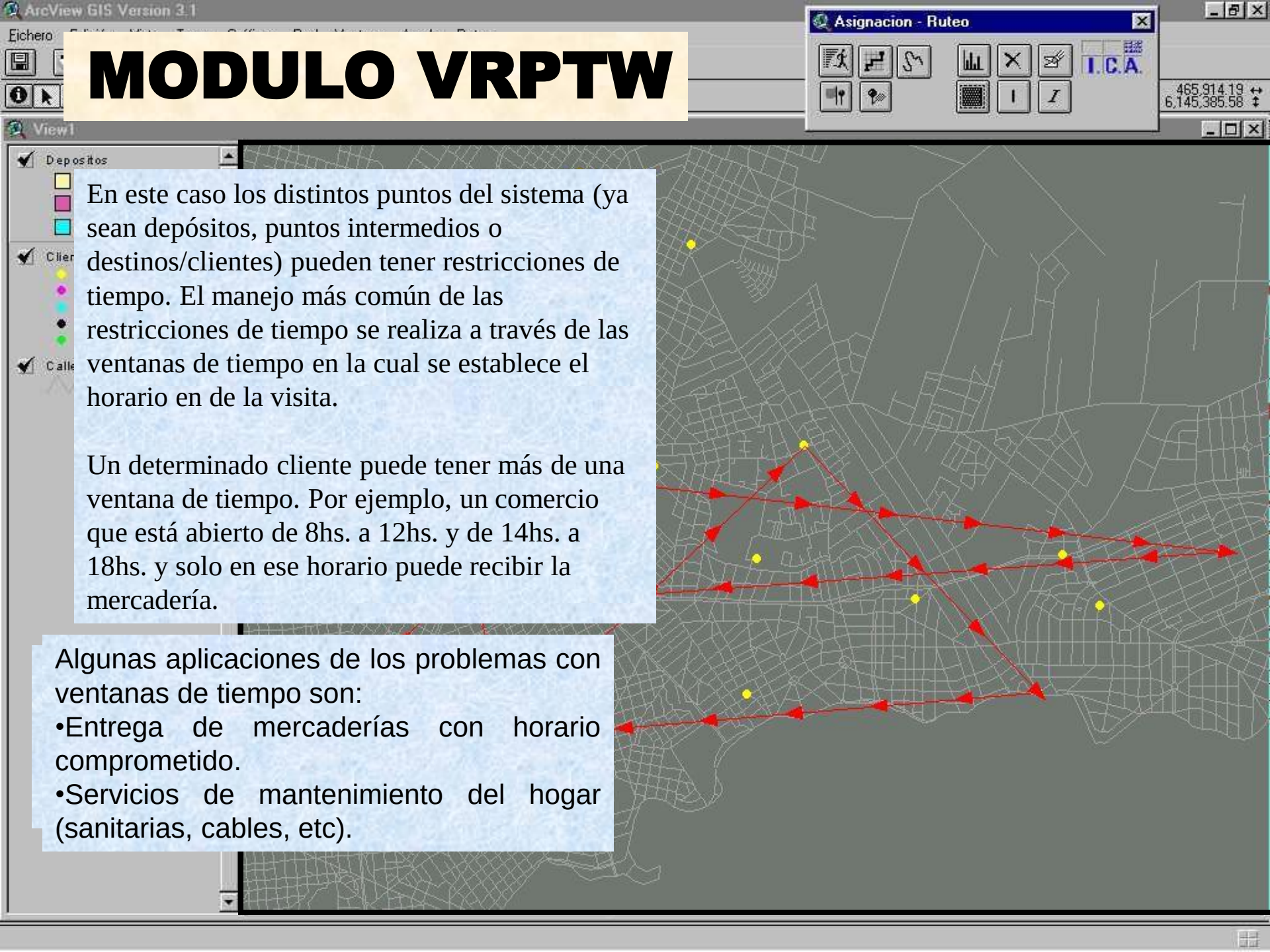
El VRP clásico es una generalización del MTSP (varios vehículos), lo cual lo hace aplicable a mayor cantidad de casos de la realidad. Las diferencias con el MTSP consisten en que en este caso los clientes tienen requerimientos de servicio que deben ser cumplidos, la flota de vehículos es limitada y los vehículos tienen capacidad limitada. Cada vehículo debe visitar un sub-conjunto de clientes que comience y termine en el depósito común, y cada cliente debe ser visitado exactamente una vez por un vehículo.



Un ejemplo de este problema es el reparto de productos de empresas que distribuyen bebidas (CoCa, Pepsi, Pilsen, Salus, etc) así como cualquier empresa que tiene a sus clientes distribuidos geográficamente. Cada cliente tiene una determinada demanda y los vehículos deben comenzar y finalizar el día en el depósito


ArcView GIS Version 3.1
Asignacion - Ruteo
I.C.A.
Ruteo
2
Clientes
Dep 0
Dep 1
Dep 2
No asignado
Dep 3
Calles.shp
Podemos ver las rutas según el orden de las visitas o sí se desea también se puede ver con más detalle las rutas que debe seguir el móvil.
Ruta detallada por donde debe circular el vehículo

MODULO VRPTW



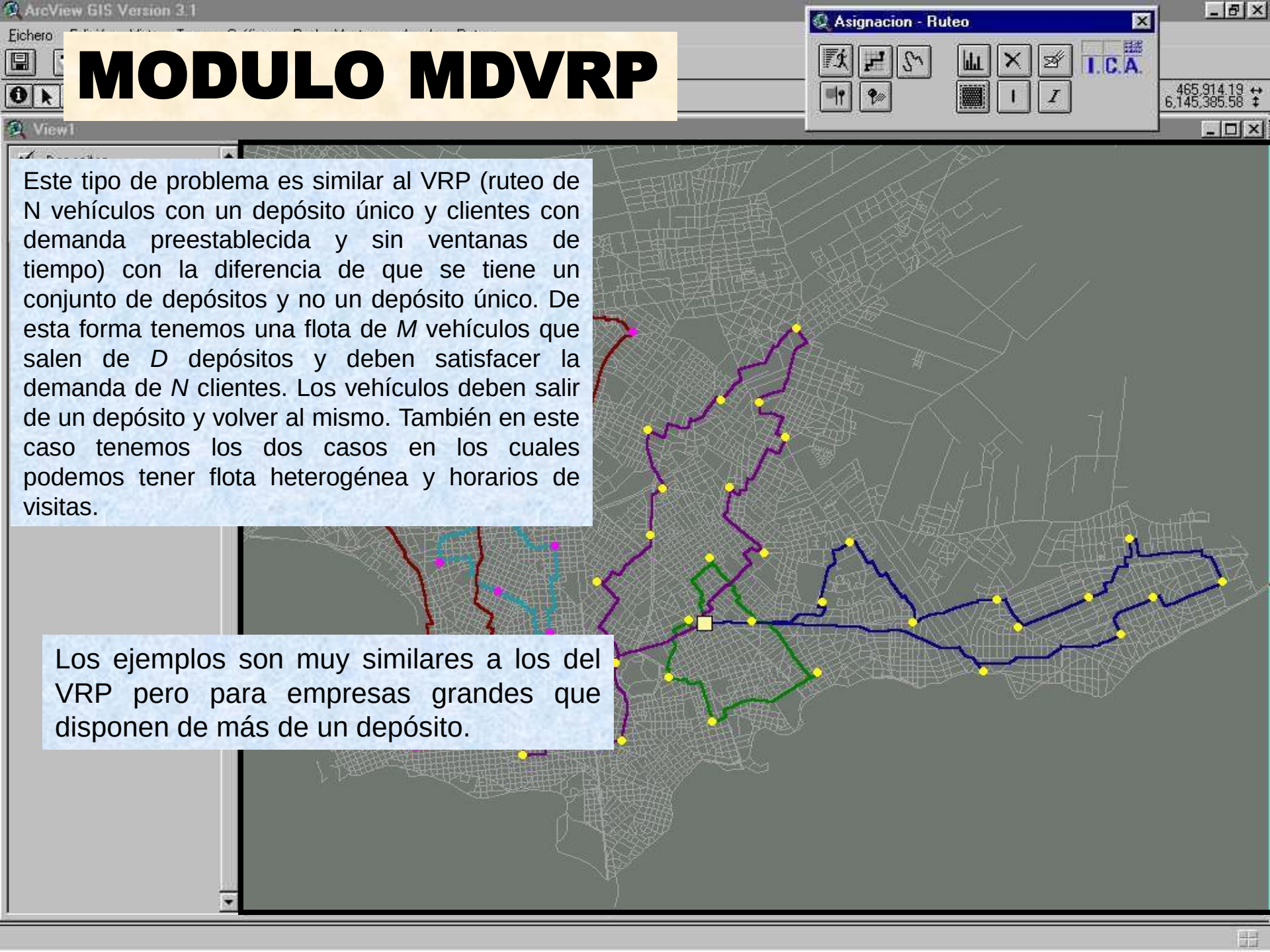
En este caso los distintos puntos del sistema (ya sean depósitos, puntos intermedios o destinos/clientes) pueden tener restricciones de tiempo. El manejo más común de las restricciones de tiempo se realiza a través de las ventanas de tiempo en la cual se establece el horario en de la visita.

Un determinado cliente puede tener más de una ventana de tiempo. Por ejemplo, un comercio que está abierto de 8hs. a 12hs. y de 14hs. a 18hs. y solo en ese horario puede recibir la mercadería.

Algunas aplicaciones de los problemas con ventanas de tiempo son:

- Entrega de mercaderías con horario comprometido.
- Servicios de mantenimiento del hogar (sanitarias, cables, etc).

MODULO MDVRP



Este tipo de problema es similar al VRP (ruteo de N vehículos con un depósito único y clientes con demanda preestablecida y sin ventanas de tiempo) con la diferencia de que se tiene un conjunto de depósitos y no un depósito único. De esta forma tenemos una flota de *M* vehículos que salen de *D* depósitos y deben satisfacer la demanda de *N* clientes. Los vehículos deben salir de un depósito y volver al mismo. También en este caso tenemos los dos casos en los cuales podemos tener flota heterogénea y horarios de visitas.

Los ejemplos son muy similares a los del VRP pero para empresas grandes que disponen de más de un depósito.

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Tiempo de Procesamiento				Secuencia de Máquinas			
	Operaciones				Operaciones		
Job	1	2	3	Job	1	2	3
j_1	3	3	2	j_1	m_1	m_2	m_3
j_2	1	5	3	j_2	m_1	m_3	m_2
j_3	3	2	3	j_3	m_2	m_1	m_3

Ejemplo de un problema de tres jobs y tres máquinas.

Objetivo: Minimizar el tiempo total de proceso (makespan)

Un schedule factible.

PENDIENTE

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

El JSSP (secuenciación de tareas) consta básicamente de un conjunto de trabajos, donde cada uno tiene un conjunto de operaciones a ser procesadas en un conjunto de recursos, llamadas máquinas.

Dichas operaciones tienen un orden y un tiempo de procesamiento en cada una de las máquinas y este no es modificable.

Cada máquina puede procesar a lo mucho un trabajo en un tiempo y una vez que un trabajo ha iniciado sobre una máquina se debe completar su procesamiento sobre esa máquina por un tiempo ininterrumpido

El objetivo es encontrar un plan de trabajo (calendario) que cumpla con las restricciones del problema y que concluya todas las operaciones de los trabajos en el menor tiempo posible (función objetivo)

Se debe miminimizar el makespan, que es el tiempo en completar todos los trabajos, es decir, la longitud del calendario desde que empieza a ejecutarse el primer trabajo hasta que finaliza el último trabajo.

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

El PJSS son combinatorios y se caracterizan por pertenecer a la clase NP-Duro,

Los problemas de planificación o programación de tareas aparecen constantemente en el mundo real.

Dichos problemas se utilizan en una amplia gama de aplicaciones diversas como:
En las líneas de producción de una fábrica.
En los hospitales para atender a los pacientes.
En los aeropuertos para despachar los vuelos.
En las escuelas para distribuir las actividades de los alumnos y profesores.
En un taller, para decidir qué equipo es reparado primero o bien la secuencia de la reparación, etc.
El PJSS es un problema que se caracteriza por pertenecer a la clase NP-Duro,

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Para probar un algoritmo que resuelva el PJSS se puede usar la familia de instancias del problema de Job Shop Scheduling conocidas como instancias de Lawrence (LA) (S. R. Lawrence,1984).
Consta de 40 problemas de 8 diferentes tamaños propuestos (M. Ventresca and B. M. Ombuki,2004):

10 x 5
15 x 5
20 x 5
10 x 10
15 x 10
20 x 10
30 x 10
15 x 15

LA (D. Applegate y W. Cook,1991) es una de las familias más comúnmente utilizadas para probar el desempeño de JSP. Cada instancia se compone de una fila de descripción y varias filas con valores enteros. Cada fila de valores enteros corresponde a un trabajo. El trabajo es un conjunto de parejas conocido como operaciones, la pareja es integrada por el número de la máquina y tiempo de procesamiento en dicha máquina.

Instancia	Tamaño
LA01	10 x 5
LA02	
LA03	
LA04	
LA05	
LA06	15 x 5
LA07	
LA08	
LA09	
LA10	
LA11	20 x 5
LA12	
LA13	
LA14	
LA15	
LA16	10x 10
LA17	
LA18	
LA19	
LA20	
LA21	15 x 10
LA22	
LA23	
LA24	
LA25	
LA26	20 x 10
LA27	
LA28	
LA29	
LA30	
LA31	30 x 10
LA32	
LA33	
LA34	
LA35	
LA36	15 x 15
LA37	
LA38	
LA39	
LA40	

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Ejemplo: en el artículo

Flórez, E., Díaz, N., Gómez, W., Bautista, L., & Delgado, D. (2018). Evaluación de algoritmos bioinspirados para la solución del problema de planificación de trabajos. *I+ D Revista de Investigaciones*, *11*(1), 142-155.

Presentan:

Un Algoritmo Colonia de Hormigas Elitista (EAS) y un algoritmo de Selección Clonal (CLONALG) para resolver el PJSS y se evalúan en las instancias de Lawrence

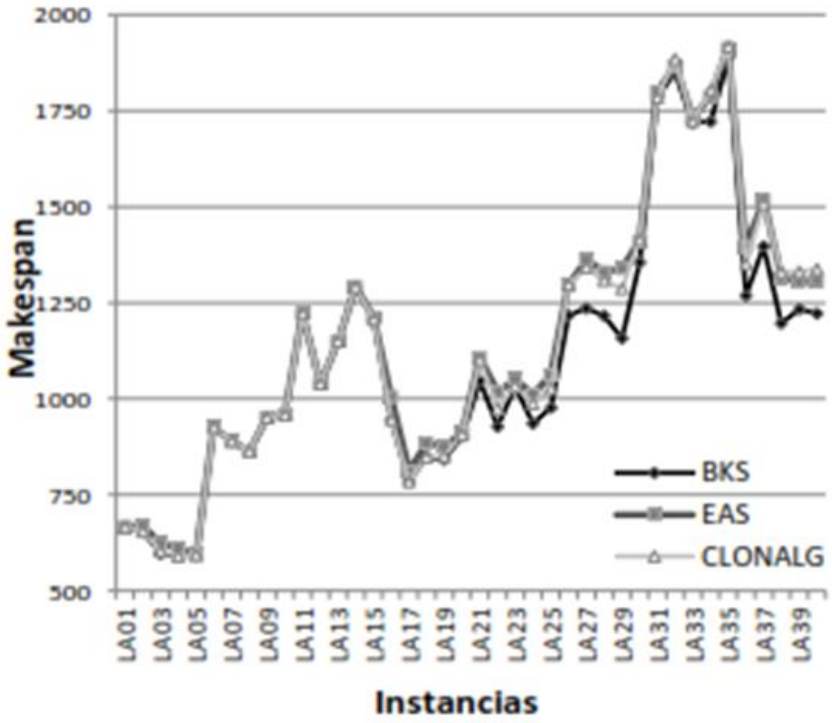
3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Se observa los resultados en cada instancia de Lawrence, se listan el mejor makespan $Cmax$ obtenido por cada algoritmo, el error relativo frente al makespan de la mejor solución conocida (Best Known Solution, BKS)

Instancia	Tamaño	BKS	EAS		CLONALG	
			C_{max}	Error relativo %	C_{max}	Error relativo %
LA01	10 x 5	**666**	**666**	0,00	**666**	0,00
LA02		**655**	669	2,14	**655**	0,00
LA03		**597**	617	3,35	603	1,01
LA04		**590**	595	0,85	**590**	0,00
LA05		**593**	**593**	0,00	**593**	0,00
LA06	15 x 5	**926**	**926**	0,00	**926**	0,00
LA07		**890**	**890**	0,00	**890**	0,00
LA08		**863**	**863**	0,00	**863**	0,00
LA09		**951**	**951**	0,00	**951**	0,00
LA10		**958**	**958**	0,00	**958**	0,00
LA11	20 x 5	**1222**	**1222**	0,00	**1222**	0,00
LA12		**1039**	**1039**	0,00	**1039**	0,00
LA13		**1150**	**1150**	0,00	**1150**	0,00
LA14		**1292**	**1292**	0,00	**1292**	0,00
LA15		**1207**	1212	0,41	**1207**	0,00
LA16	10x 10	**945**	996	5,40	946	0,11
LA17		**784**	812	3,57	**784**	0,00
LA18		**848**	885	4,36	**848**	0,00
LA19		**842**	873	3,68	851	1,07
LA20		**902**	912	1,11	907	0,55
LA21	15 x 10	**1046**	1107	5,83	1102	5,35
LA22		**927**	995	7,34	974	5,07
LA23		**1032**	1049	1,65	1033	0,10
LA24		**935**	1008	7,81	987	5,56
LA25		**977**	1062	8,70	1028	5,22
LA26	20 x 10	**1218**	1296	6,40	1297	6,49
LA27		**1235**	1349	9,23	1342	8,66
LA28		**1216**	1322	8,72	1308	7,57
LA29		**1157**	1331	15,04	1286	11,15
LA30		**1355**	1410	4,06	1414	4,35
LA31	30 x 10	**1784**	**1784**	0,00	**1784**	0,00
LA32		**1850**	1860	0,54	1884	1,84
LA33		**1719**	1731	0,70	1723	0,23
LA34		**1721**	1778	3,31	1804	4,82
LA35		**1888**	1902	0,74	1918	1,59
LA36	15 x 15	**1268**	1396	10,09	1352	6,62
LA37		**1397**	1517	8,59	1508	7,95
LA38		**1196**	1315	9,95	1330	11,20
LA39		**1233**	1304	5,76	1331	7,95
LA40		**1222**	1300	6,38	1338	9,49

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Comportamiento de los Makespan de los algoritmos para el JSP (Best Known Solution [BKS], Algoritmo Colonia de Hormigas Elitista [EAS], Algoritmo de Selección Clonal [CLONALG])

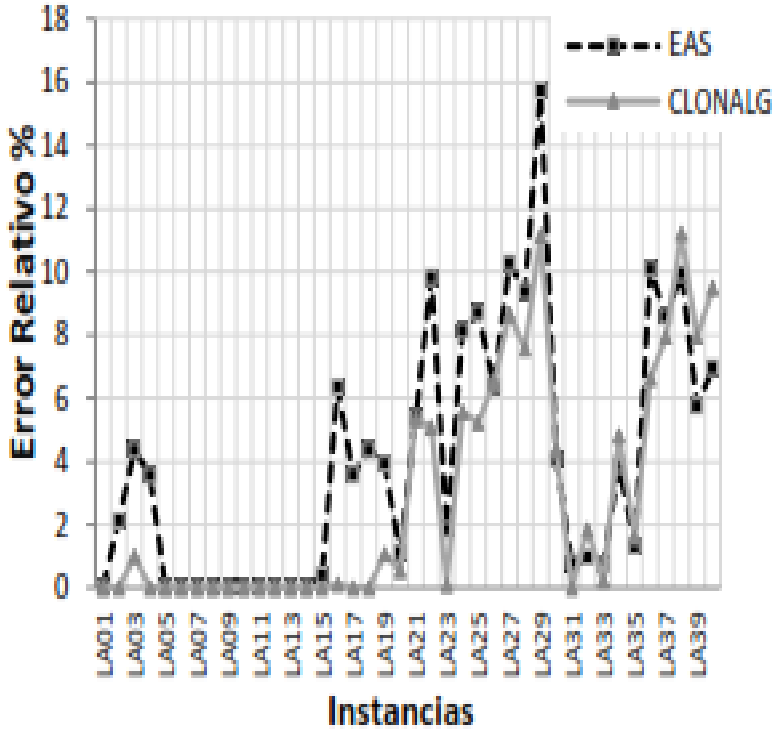


Instancia	Tamaño	BKS	EAS		CLONALG	
			C_{max}	Error relativo %	C_{max}	Error relativo %
LA01	10 x 5	**666**	**666**	0,00	**666**	0,00
LA02		**655**	669	2,14	**655**	0,00
LA03		**597**	617	3,35	603	1,01
LA04		**590**	595	0,85	**590**	0,00
LA05		**593**	**593**	0,00	**593**	0,00
LA06	15 x 5	**926**	**926**	0,00	**926**	0,00
LA07		**890**	**890**	0,00	**890**	0,00
LA08		**863**	**863**	0,00	**863**	0,00
LA09		**951**	**951**	0,00	**951**	0,00
LA10		**958**	**958**	0,00	**958**	0,00
LA11	20 x 5	**1222**	**1222**	0,00	**1222**	0,00
LA12		**1039**	**1039**	0,00	**1039**	0,00
LA13		**1150**	**1150**	0,00	**1150**	0,00
LA14		**1292**	**1292**	0,00	**1292**	0,00
LA15		**1207**	1212	0,41	**1207**	0,00
LA16	10x 10	**945**	996	5,40	946	0,11
LA17		**784**	812	3,57	**784**	0,00
LA18		**848**	885	4,36	**848**	0,00
LA19		**842**	873	3,68	851	1,07
LA20		**902**	912	1,11	907	0,55
LA21	15 x 10	**1046**	1107	5,83	1102	5,35
LA22		**927**	995	7,34	974	5,07
LA23		**1032**	1049	1,65	1033	0,10
LA24		**935**	1008	7,81	987	5,56
LA25		**977**	1062	8,70	1028	5,22
LA26	20 x 10	**1218**	1296	6,40	1297	6,49
LA27		**1235**	1349	9,23	1342	8,66
LA28		**1216**	1322	8,72	1308	7,57
LA29		**1157**	1331	15,04	1286	11,15
LA30		**1355**	1410	4,06	1414	4,35
LA31	30 x 10	**1784**	**1784**	0,00	**1784**	0,00
LA32		**1850**	1860	0,54	1884	1,84
LA33		**1719**	1731	0,70	1723	0,23
LA34		**1721**	1778	3,31	1804	4,82
LA35		**1888**	1902	0,74	1918	1,59
LA36	15 x 15	**1268**	1396	10,09	1352	6,62
LA37		**1397**	1517	8,59	1508	7,95
LA38		**1196**	1315	9,95	1330	11,20
LA39		**1233**	1304	5,76	1331	7,95
LA40		**1222**	1300	6,38	1338	9,49

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

Comportamiento del error relativo para las instancias de Lawrence (Algoritmo Colonia de Hormigas Elitista [EAS], Algoritmo de Selección Clonal [CLONALG])

Se observa que conforme las instancias son más grandes y de mayor complejidad los errores crecen en la mayoría de los casos haciendo una excepción en la instancia 23 y el rango de la 30 a la 35, este comportamiento es similar en los 2 algoritmos.

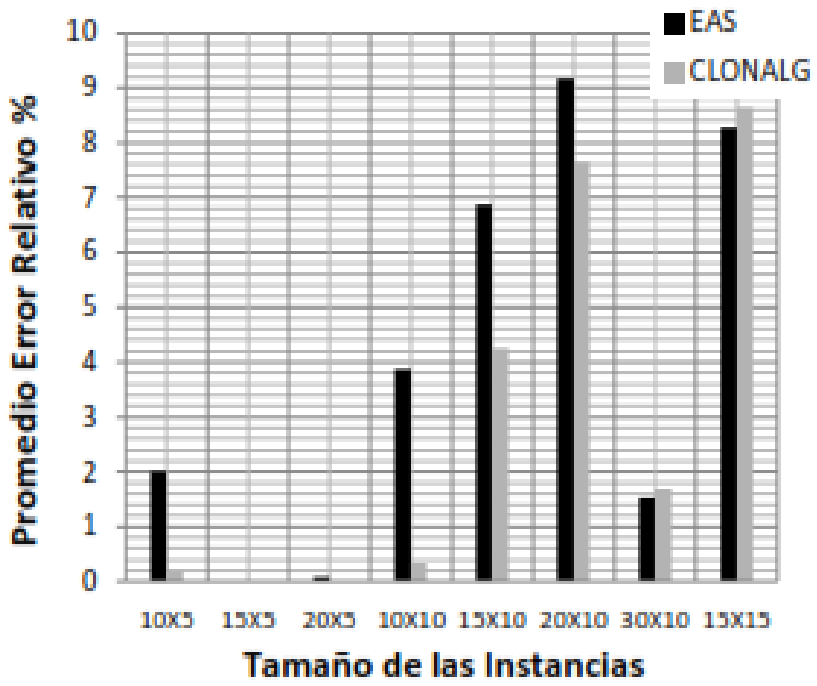


Instancia	Tamaño	BKS	EAS		CLONALG	
			C_{max}	Error relativo %	C_{max}	Error relativo %
LA01	10 x 5	**666**	**666**	0,00	**666**	0,00
LA02		**655**	669	2,14	**655**	0,00
LA03		**597**	617	3,35	603	1,01
LA04		**590**	595	0,85	**590**	0,00
LA05		**593**	**593**	0,00	**593**	0,00
LA06	15 x 5	**926**	**926**	0,00	**926**	0,00
LA07		**890**	**890**	0,00	**890**	0,00
LA08		**863**	**863**	0,00	**863**	0,00
LA09		**951**	**951**	0,00	**951**	0,00
LA10		**958**	**958**	0,00	**958**	0,00
LA11	20 x 5	**1222**	**1222**	0,00	**1222**	0,00
LA12		**1039**	**1039**	0,00	**1039**	0,00
LA13		**1150**	**1150**	0,00	**1150**	0,00
LA14		**1292**	**1292**	0,00	**1292**	0,00
LA15		**1207**	1212	0,41	**1207**	0,00
LA16	10x 10	**945**	996	5,40	946	0,11
LA17		**784**	812	3,57	**784**	0,00
LA18		**848**	885	4,36	**848**	0,00
LA19		**842**	873	3,68	851	1,07
LA20		**902**	912	1,11	907	0,55
LA21	15 x 10	**1046**	1107	5,83	1102	5,35
LA22		**927**	995	7,34	974	5,07
LA23		**1032**	1049	1,65	1033	0,10
LA24		**935**	1008	7,81	987	5,56
LA25		**977**	1062	8,70	1028	5,22
LA26	20 x 10	**1218**	1296	6,40	1297	6,49
LA27		**1235**	1349	9,23	1342	8,66
LA28		**1216**	1322	8,72	1308	7,57
LA29		**1157**	1331	15,04	1286	11,15
LA30		**1355**	1410	4,06	1414	4,35
LA31	30 x 10	**1784**	**1784**	0,00	**1784**	0,00
LA32		**1850**	1860	0,54	1884	1,84
LA33		**1719**	1731	0,70	1723	0,23
LA34		**1721**	1778	3,31	1804	4,82
LA35		**1888**	1902	0,74	1918	1,59
LA36	15 x 15	**1268**	1396	10,09	1352	6,62
LA37		**1397**	1517	8,59	1508	7,95
LA38		**1196**	1315	9,95	1330	11,20
LA39		**1233**	1304	5,76	1331	7,95
LA40		**1222**	1300	6,38	1338	9,49

3.- EL PROBLEMA DEL JOB SHOP SCHEDULING (PJSS)

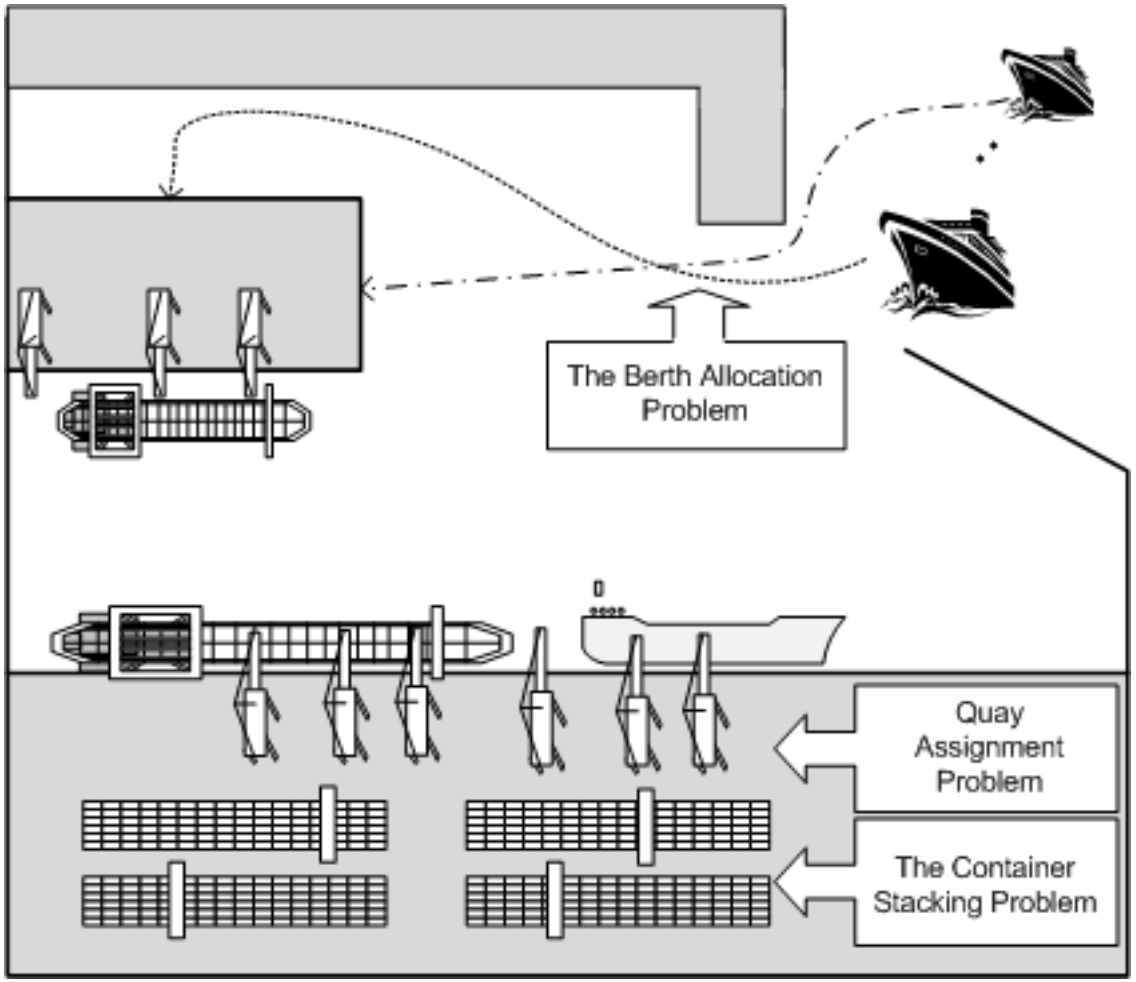
Representación del error relativo vs tamaño de las instancias para cada algoritmo en el JSP (Algoritmo Colonia de Hormigas Elitista [EAS], Algoritmo de Selección Clonal [CLONALG])

Es interesante observar que aunque en la mayoría de tamaños el mejor rendimiento lo obtuvo el algoritmo CLONALG para el caso de las instancias de mayor tamaño (30x10 y 15X15) presenta un mejor desempeño el algoritmo EAS.



Instancia	Tamaño	BKS	EAS		CLONALG	
			C_{max}	Error relativo %	C_{max}	Error relativo %
LA01	10 x 5	**666**	**666**	0,00	**666**	0,00
LA02		**655**	669	2,14	**655**	0,00
LA03		**597**	617	3,35	603	1,01
LA04		**590**	595	0,85	**590**	0,00
LA05		**593**	**593**	0,00	**593**	0,00
LA06	15 x 5	**926**	**926**	0,00	**926**	0,00
LA07		**890**	**890**	0,00	**890**	0,00
LA08		**863**	**863**	0,00	**863**	0,00
LA09		**951**	**951**	0,00	**951**	0,00
LA10		**958**	**958**	0,00	**958**	0,00
LA11	20 x 5	**1222**	**1222**	0,00	**1222**	0,00
LA12		**1039**	**1039**	0,00	**1039**	0,00
LA13		**1150**	**1150**	0,00	**1150**	0,00
LA14		**1292**	**1292**	0,00	**1292**	0,00
LA15		**1207**	1212	0,41	**1207**	0,00
LA16	10x 10	**945**	996	5,40	946	0,11
LA17		**784**	812	3,57	**784**	0,00
LA18		**848**	885	4,36	**848**	0,00
LA19		**842**	873	3,68	851	1,07
LA20		**902**	912	1,11	907	0,55
LA21	15 x 10	**1046**	1107	5,83	1102	5,35
LA22		**927**	995	7,34	974	5,07
LA23		**1032**	1049	1,65	1033	0,10
LA24		**935**	1008	7,81	987	5,56
LA25		**977**	1062	8,70	1028	5,22
LA26	20 x 10	**1218**	1296	6,40	1297	6,49
LA27		**1235**	1349	9,23	1342	8,66
LA28		**1216**	1322	8,72	1308	7,57
LA29		**1157**	1331	15,04	1286	11,15
LA30		**1355**	1410	4,06	1414	4,35
LA31	30 x 10	**1784**	**1784**	0,00	**1784**	0,00
LA32		**1850**	1860	0,54	1884	1,84
LA33		**1719**	1731	0,70	1723	0,23
LA34		**1721**	1778	3,31	1804	4,82
LA35		**1888**	1902	0,74	1918	1,59
LA36	15 x 15	**1268**	1396	10,09	1352	6,62
LA37		**1397**	1517	8,59	1508	7,95
LA38		**1196**	1315	9,95	1330	11,20
LA39		**1233**	1304	5,76	1331	7,95
LA40		**1222**	1300	6,38	1338	9,49

4.- PROBLEMA DE ATRAQUE DE BARCOS
BERTH ALLOCATION PROBLEM (BAP)

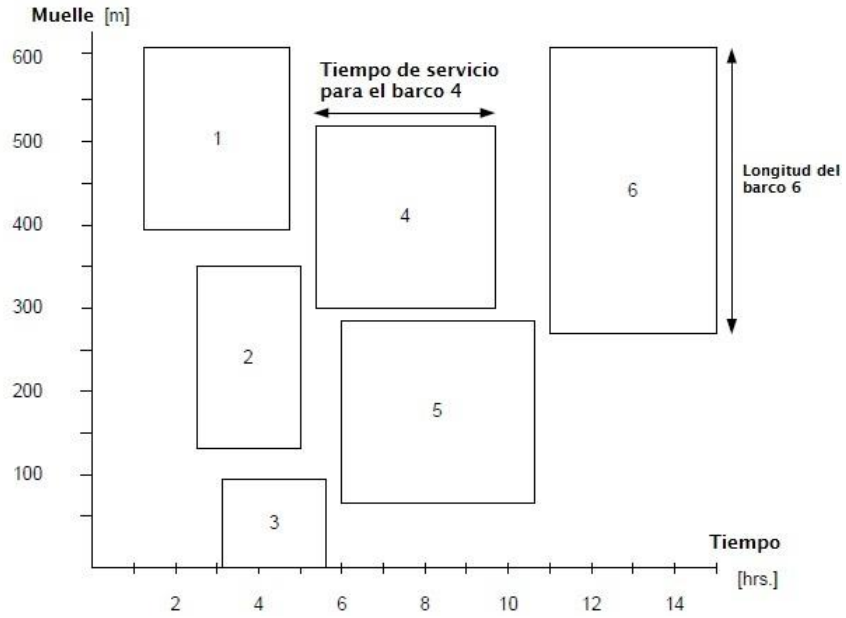


PENDIENTE

4.- PROBLEMA DE ATRAQUE DE BARCOS
BERTH ALLOCATION PROBLEM (BAP)

4.- PROBLEMA DE ATRAQUE DE BARCOS
BERTH ALLOCATION PROBLEM (BAP)

BERTH ALLOCATION PROBLEM: qué espacio en el muelle y qué tiempo de servicio asignar a los barcos que han de cargarse y descargarse en una terminal.



- El problema es NP-duro.

[Lim1998]

DATOS DE ENTRADA

Longitudes de los barcos
Tiempos de llegada
Tiempos de trabajo
Tipo de muelle

PROBLEMA

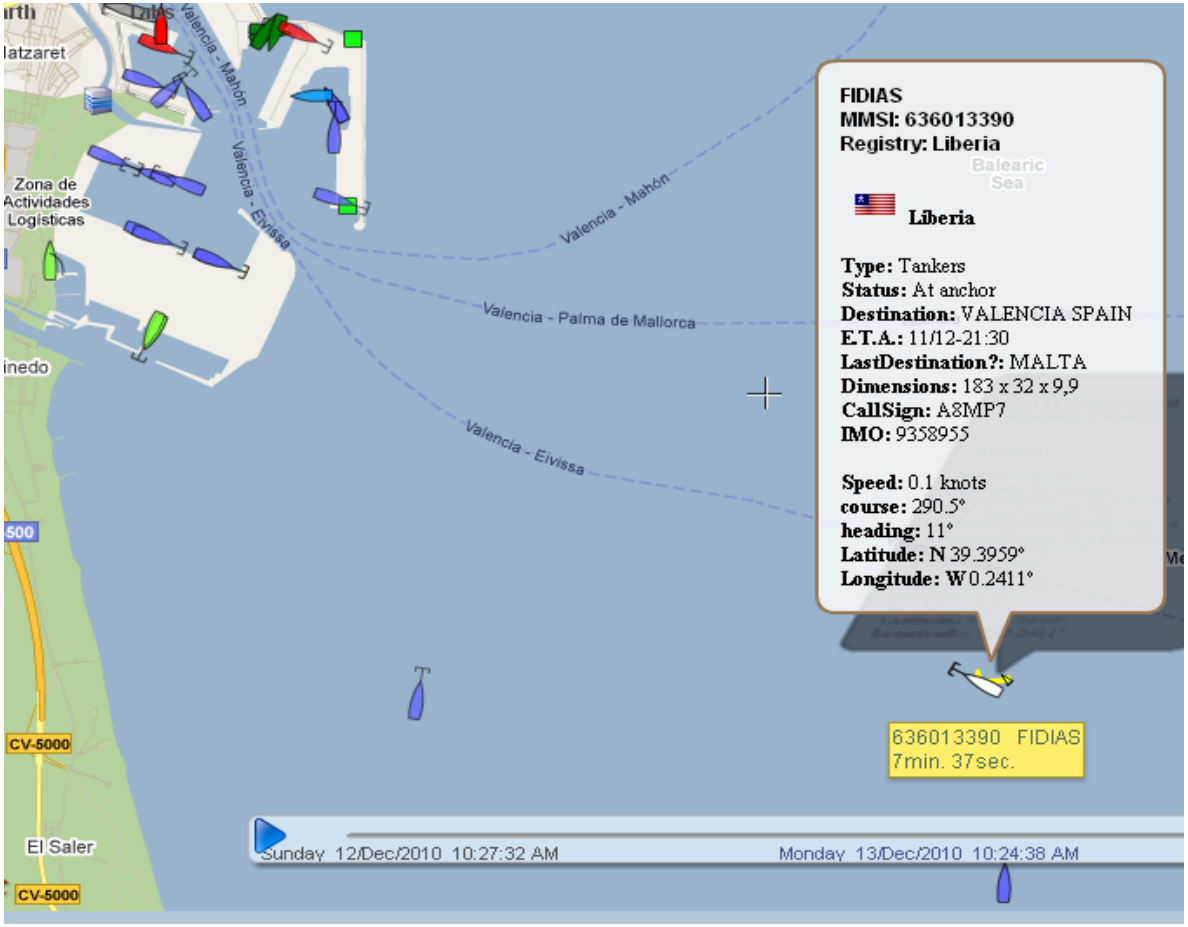
BERTH ALLOCATION PROBLEM

DATOS DE SALIDA

PLAN DE ATRAQUE

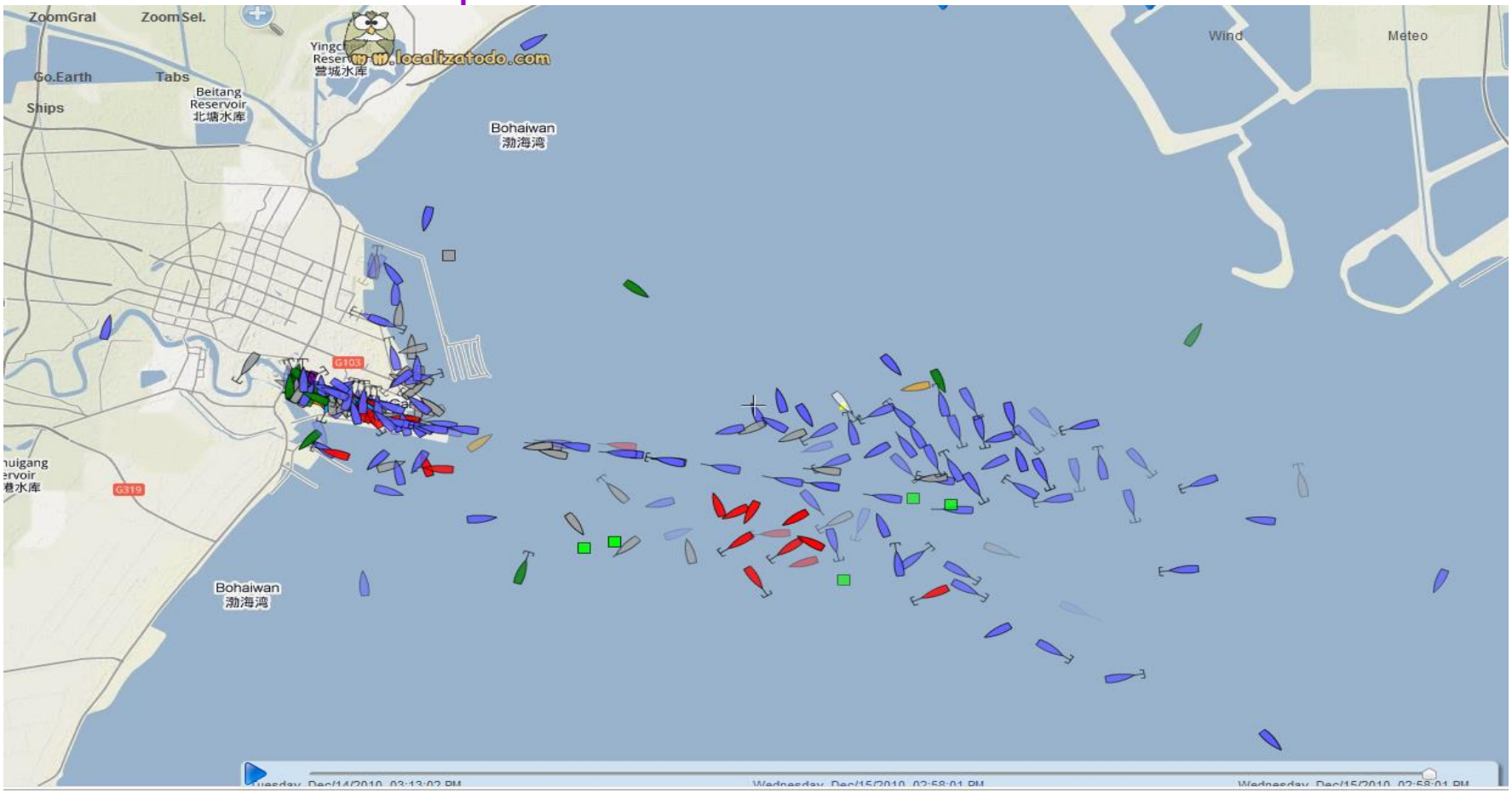
4.- PROBLEMA DE ATRAQUE DE BARCOS
BERTH ALLOCATION PROBLEM (BAP)

http://www.localizatodo.com/
http://www.marinetraffic.com/



4.- PROBLEMA DE ATRAQUE DE BARCOS
BERTH ALLOCATION PROBLEM (BAP)

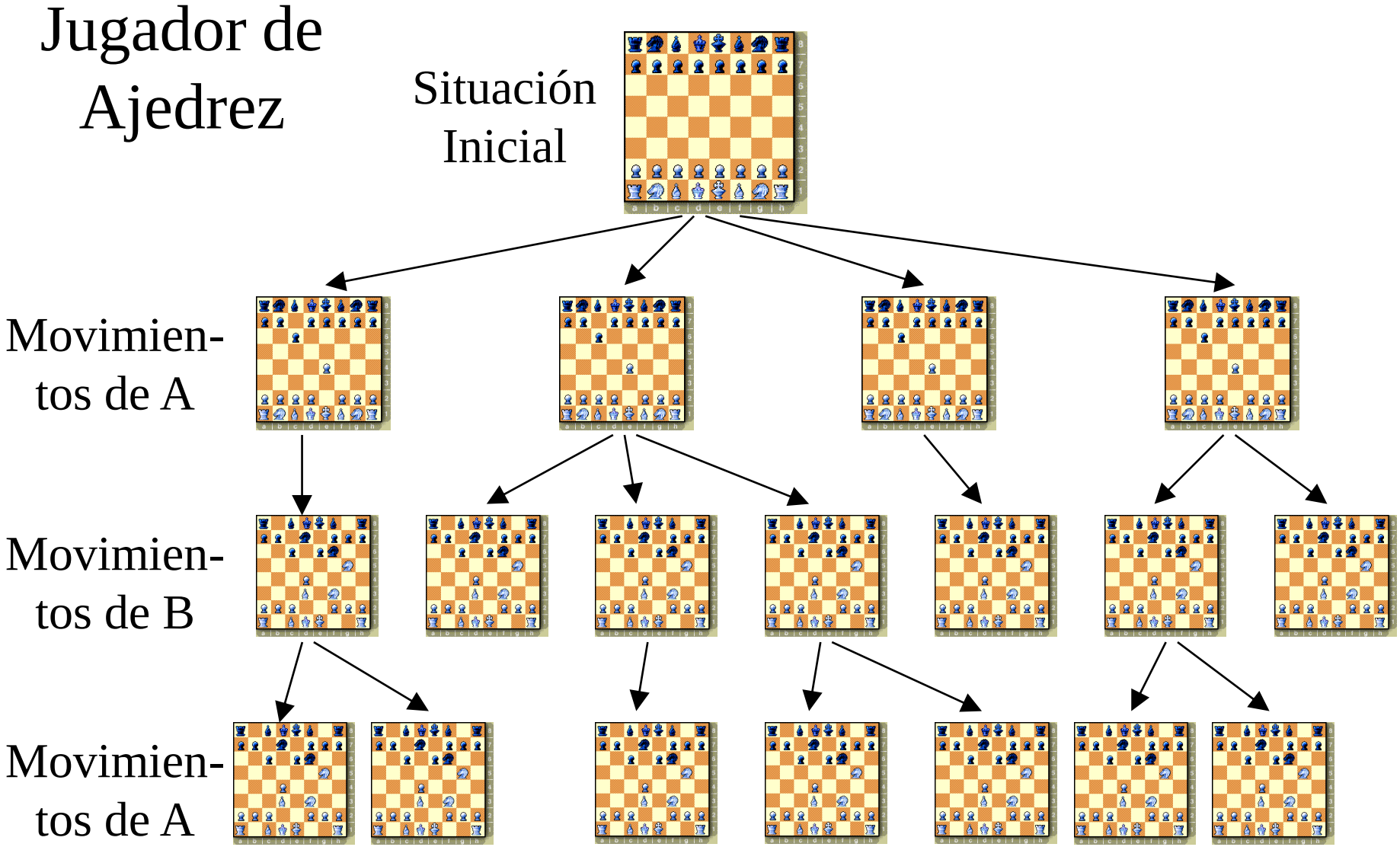
http://www.localizatodo.com/
http://www.marinetraffic.com/



5 .- JUEGO DEL AJEDREZ

Encontrar un algoritmo que permita siempre ganar la partida de

5 .- JUEGO DEL AJEDREZ



5 .- JUEGO DEL AJEDREZ

Para juegos complejos, como el ajedrez, el árbol de búsqueda adquiere dimensiones inimaginables.. Si suponemos que un computador tarda 1/3 de nanosegundo (15^{-9} seg.) en generar cada sucesor, el árbol del ajedrez sería generado en:

$10^{120}*15*10^{(-9)}$ seg.

=150000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
seg.

= aproximadamente 1021 siglos.

Por consiguiente, por ahora se debe desechar la idea de generar en juegos complejos todo el árbol de búsqueda con la intención de determinar de antemano una estrategia ganadora.

Cita...

- “No hay un incremento concebible en el poder de las computadoras que pueda saturar la demanda científica: aún pensando que una computadora posea un ciclo de tiempo subnuclear (10^{-23} seg.) y densidades de almacenamiento subnucleares (10^{39} bits/cm^3), ésta no podría manejar la mayoría de los problemas que son importantes en la investigación científica básica y aplicada. Por lo tanto, existirá siempre una fuerte presión para incrementar la **eficiencia** de los programas, para poder incrementar también la cantidad de información últil generada por un programa.”

  Ken Wilson, Nóbel de Física 1982