

Estudio del Espacio de Soluciones del Problema del Cajero Viajante

Pedro Esteban Gardel Sotomayor

Universidad Nacional de Asunción

Centro Nacional de Computación

San Lorenzo, Paraguay

pgardel@cnc.una.py

Osvaldo Gómez

Universidad Nacional de Asunción

Centro Nacional de Computación

San Lorenzo, Paraguay

ogomez@cnc.una.py

Benjamín Barán

Universidad Nacional de Asunción

Centro Nacional de Computación

San Lorenzo, Paraguay

bbaran@cnc.una.py

Resumen

El presente trabajo estudia el espacio de soluciones del Problema del Cajero Viajante o el *Traveling Salesman Problem* (*TSP*). Debido al enorme tamaño del espacio de soluciones de los problemas estudiados, se ha decidido tomar muestras de los mismos con el objetivo de tener una visión general de su estructura. Para este fin, se utilizaron dos políticas en la obtención de las muestras, una basada en estudios anteriores y otra aquí propuesta por los autores. Se han tomado veinte instancias del *TSP* de la *TSPLIB*. El análisis de los resultados obtenidos es congruente en todos los casos con la conjectura de un espacio de soluciones globalmente convexo del *TSP*. Esto es, un espacio con características de “*Gran Valle*” como fuera sugerido por Boese, y verificado experimentalmente en el presente trabajo.

Palabras Claves: *Traveling Salesman Problem*, Optimización local, Espacio globalmente convexo.

Abstract

The present paper studies the space of solutions of the Traveling Salesman Problem (*TSP*). Due to the enormous size of the solutions space of the studied problems, it has been decided to take samples with the objective of having a general vision of the problem structure. For this purpose, two policies were used to obtain the samples, one already used in previous studies and the other designed by the authors. For this study, 20 instances of the *TSP* have been taken from the *TSPLIB*. The analysis of the results was coherent with the conjecture of a globally convex structure of the *TSP*’s solutions space. This is a space that has characteristics of “*Great Valley*” as it was suggested by Boese and experimentally proved in the present work.

Keywords: Traveling Salesman Problem, Local optimization, Globally convex space.

1 Introducción

El Problema del Cajero Viajante, más conocido por sus siglas en inglés *TSP* (*Traveling Salesman Problem*), ha sido utilizado como “*Problema de Prueba*” (o problema *paradigma*) para estudiar el comportamiento de muchos algoritmos de optimización [1,2,4,5]. Por lo tanto resulta interesante un estudio más extenso del espacio de soluciones del mismo con el fin de conocer su estructura. Este trabajo se inspira en las propuestas de Boese [1,2] y otros autores como Stützle et al. [5] que han realizado algunos estudios similares para instancias específicas del *TSP*, pero no de manera sistemática, como se propone en el presente trabajo. Estos estudios consistieron en general en hallar óptimos locales mediante distintos algoritmos de búsqueda local, para luego analizar los resultados obtenidos e intentar entender la estructura del *TSP* para las instancias estudiadas.

En contrapartida a los trabajos citados, el presente trabajo analiza un conjunto de 20 problemas de solución óptima conocida, extraídos de la *TSPLIB* [6], una fuente de problemas referenciales del *TSP*, muy utilizada por diversos investigadores del área. El presente trabajo propone entonces obtener soluciones factibles del espacio de búsqueda (o muestras), compararlas con la solución óptima y con las demás soluciones factibles, para finalmente intentar conocer la estructura del *TSP*. Para obtener estas muestras, se utilizaron básicamente 2 políticas de muestreo:

1. Siguiendo los trabajos de Boese [1,2], se utilizó una heurística de optimización local, conocida como *3-opt*, con iniciación aleatoria.
2. Alternativamente, se generaron muestras de soluciones uniformemente distribuidas en el espacio de soluciones, para cada problema en estudio.

El trabajo fue organizado de la siguiente manera: a continuación se describe el *TSP* y se formaliza la notación a ser utilizada. Seguidamente, se describen los métodos utilizados para obtener las muestras de soluciones del *TSP*. En la cuarta sección se presentan los resultados experimentales mientras que en la sección 5 se describe la estructura del *TSP* como un espacio globalmente convexo. Finalmente se presentan las conclusiones del estudio.

2 Problema del Cajero Viajante

El *TSP* puede ser representado por un conjunto de ciudades o nodos $V=\{N1, N2, \dots, Nn\}$ con n elementos (o ciudades) y un conjunto A de arcos que asocia a cada par de ciudades o nodos $\{Ni, Nj\}$ una distancia $d(i,j)$. La solución del *TSP* es conocida como *tour* (r) y está dada por un conjunto ordenado de las n ciudades o nodos $\{Ni\}$ que contiene a todas las ciudades del problema exactamente una vez, esto es:

$$r = \{N(r(1)), N(r(2)), \dots, N(r(n))\}; \quad N(r(i)) \in V, \quad N(r(i)) \neq N(r(j)) \text{ si } i \neq j \quad (1)$$

Cada *tour* tiene asociado una longitud definida como:

$$l(r) = \sum_{i=1}^{n-1} d(N(r(i)), N(r(i+1))) + d(N(r(n)), N(r(1))). \quad (2)$$

El objetivo del *TSP* es hallar al *tour* de menor longitud, también conocido como *tour* óptimo (r^*).

El presente trabajo sólo considera problemas en los que los arcos representen distancias euclidianas entre los nodos i y j , por lo tanto, $d(i,j) = d(j,i)$. Este tipo de problemas es conocido como *TSP simétrico*. En este contexto, dados dos *tours* $r1$ y $r2$, la distancia $\delta(r1, r2)$ se define como n menos la cantidad de arcos comunes a ambos *tours*. Lógicamente, la distancia de un *tour* r a la solución óptima r^* estará denotada como $\delta(r, r^*)$.

Una población P es un conjunto de m soluciones o *tours*, esto es:

$$\text{Población: } P = \{r1, r2, \dots, rm\} \quad (3)$$

La distancia media de un *tour* r a una población P se define como

$$\delta(P, r) = \frac{1}{m} \sum_{i=1}^m \delta(r_i, r) \quad (4)$$

y da una idea de cuan cerca está un *tour* a una población P . Corresponde hacer notar que dada una población P uniformemente distribuida en un sub-espacio, los *tours* que se encuentren en su centro tendrán una menor distancia media a la población que los que se encuentren en su periferia, por lo que $\delta(P, r)$ podrá ser utilizado para estimar la posición de un punto respecto a una población.

3 Métodos de muestreo utilizados para el análisis

Inspirado en los trabajos de Boese [1,2], el primer método utilizado intenta crear una población de soluciones localmente óptimas, para lo que se utilizará un algoritmo que se dará en llamar “*3-opt*”. El mismo utiliza una heurística de búsqueda local mostrada en el *Pseudocódigo 1*, para hallar m óptimos locales y conformar así una población de estudio P . Este acercamiento ya fue abordado por otros autores en [1,4,5] demostrándose que las

soluciones obtenidas se concentran en una zona determinada del espacio de soluciones y no se distribuyen uniformemente en todo el espacio.

A raíz de la distribución no uniforme de la población de óptimos locales en publicaciones anteriores, este trabajo propone adicionalmente el estudio de una población más uniforme, con soluciones distribuidas a lo largo de un mayor espacio, para lo cual se propone un segundo método de selección de muestras. Este método, que damos en llamar “*Dist*”, obtiene un conjunto de soluciones con la característica de tener un subconjunto de cardinalidad constante, perteneciente a cada posible distancia del óptimo r^* .

3.1 Descripción del algoritmo 3-opt

Al inicio, esta heurística guarda en una variable r_{opt} un *tour* hallado aleatoriamente de la siguiente manera: el *tour* comienza en una ciudad cualquiera del problema, luego se elige aleatoriamente otra ciudad de V , siendo V el conjunto de ciudades todavía no visitas, y se elimina ésta de V . El proceso se repite iterativamente hasta que el conjunto V quede vacío y se hayan visitado todas las ciudades del problema, por lo que solo queda regresar a la ciudad de origen.

Una vez encontrado un *tour* inicial, se realiza iterativamente el siguiente proceso: Se cambian tres arcos elegidos al azar de r_{opt} , dando origen a un nuevo *tour* válido que se guarda en r_{new} . Se comparan las longitudes de r_{opt} y r_{new} , si $l(r_{new})$ es menor, se guarda r_{new} en r_{opt} . Se realiza este proceso hasta que r_{opt} no varíe durante una cantidad de iteraciones (CI) igual a 1000 para los resultados experimentales presentados en este trabajo.

El proceso mencionado se repite hasta obtener el Número de *Tours* (NT) deseado para generar la población P . Para este trabajo, se escogió $NT = 1000$ *tours* y se calcularon los correspondientes valores de longitud $l(r)$, distancia al óptimo $\delta(r, r^*)$ y distancia media a la población $\delta(P, r)$, para cada *tour* $r \in P$. A continuación se presenta el pseudocódigo de 3-opt, donde los comentarios van precedidos por el símbolo %.

Pseudocódigo 1: 3-opt

Parámetros de entrada: matriz de distancias $D = d(i, j)$, número de *tours* NT , cantidad de iteraciones CI

Parámetros de salida: población P , con los correspondientes valores de $l(r)$, $\delta(r, r^*)$ y $\delta(P, r)$

Iniciarizar $k = 0$

Repetir mientras $k < NT$

$r_{opt} = \text{Nuevo tour}()$ % Se halla un recorrido en forma aleatoria (r_{opt}).

 Hallar $l(r_{opt})$

 Iniciarizar $c = 0$

Repetir mientras $c < CI$

$r_{new} = \text{Cambios}(r_{opt})$ % Se realizan los cambios en r_{opt} para generar r_{new}

 Hallar $l(r_{new})$

Si $l(r_{new}) < l(r_{opt})$ % Se comparan las longitudes de r_{new} y r_{opt}

$r_{opt} = r_{new}$ % Si $l(r_{new})$ es menor se cambia r_{opt} por r_{new}

$l(r_{opt}) = l(r_{new})$

$c = 0$

Fin si

$c = c + 1$

Fin mientras

$P(k) = r_{opt}$

 Calcular $\delta(r_k, r^*)$ % Se calcula la distancia al óptimo del nuevo *tour* r_k .

 Calcular $l(r_k)$

$k = k + 1$

Fin mientras

Para todo *tour* $r \in P$.

 Calcular $\delta(P, r)$ % Se calcula la distancia media a la población

Fin para. % de cada individuo de P .

Función $r_{opt} = \text{Nuevo tour}()$

$r_{opt} = \text{conjunto vacío}$

Mientras V no sea vacío

 Se escoge al azar una ciudad (Ni) de V

 Se elimina Ni de V

$r_{opt} = (r_{opt}, Ni)$

Fin mientras

Función $r_{new} = Cambios (r_{opt})$

Seleccionar tres arcos al azar de r_{opt}

Cortar r_{opt} en los arcos seleccionados

Reordenar los segmentos de r_{opt} para construir un nuevo *tour* (r_{new}) con $\delta(r_{new}, r_{opt}) = 3$

3.2 Descripción del algoritmo *Dist*

Este algoritmo tiene como objetivo hallar un número de *tours* (NT') a todas las distancias posibles del óptimo r^* . Las distancias posibles al óptimo van de 2 hasta n , siendo n la cantidad de ciudades en el problema. Para cada una de estas distancias se genera entonces un subconjunto de NT' soluciones, por lo que el número total de *tours* estudiados es de: $NT'(n-1)$. Por ejemplo, para un problema de 280 ciudades (como el *a280* mostrado en la tabla 1), con $NT'=24$, se analizaron 6696 *tours* componentes de la población P , mientras que para el problema *pr1002* con $NT'=6$ (ver Tabla 1) la población P contenía solo 6006 *tours*.

Este algoritmo selecciona arcos al azar en el *tour* óptimo r^* y los reordena formando nuevo *tour* válido r_{new} . La cantidad de arcos seleccionados depende de la distancia al óptimo a la que se desea tomar muestras. Seguidamente se calcula $\delta(r^*, r_{new})$ y se comprueba si r_{new} está a la distancia deseada, de no ser así, se continua el proceso de cambio de arcos. Una vez que se han obtenido la cantidad NT' de *tours* deseados a una distancia dada del óptimo, se procede a hallar las siguientes muestras, posiblemente a la distancia siguiente. En consecuencia, el algoritmo también produce una población P de *tours*, con sus correspondientes valores de $l(r)$, $\delta(r, r^*)$ y $\delta(P, r)$. A continuación se describe el pseudocódigo correspondiente.

Pseudocódigo 2: *Dist*

Parámetros de entrada: r^* , número de *tours* NT' , matriz de distancias $D = d(i,j)$.

Parámetros de salida: población P , con los correspondientes valores de $l(r)$, $\delta(r, r^*)$ y $\delta(P, r)$

n = cantidad de ciudades en r^* (es la misma cantidad de ciudades del problema)

Inicializar P como conjuntos vacíos

Para $nc = 2$ hasta n

Para $k = 1$ hasta NT'

$r_{new} = Cambios (r^*, nc)$

 Hallar $l(r_{new})$

$P = [P, r_{new}]$

$l(r_k) = l(r_{new})$

$\delta(r_k, r^*) = nc$.

 % Se hallan los $NT'(n-1)$ *tours* que se quieren.

 % Se hallan NT' *tours* a la distancia nc del óptimo

Fin para

Fin para

Para cada $r \in P$

 Calcular $\delta(P, r)$

 % La distancia al óptimo se determinó en *Cambios*

Fin para.

Función *Cambios* (r^*, nc)

$r_{new} = r^*$

Salida = 0

Mientras Salida = 0

 Seleccionar arcos al azar de r_{new} y cortar el *tour*

 Reconstruir r_{new} con los segmentos

 y obtener un nuevo *tour* válido: r_{new}

 Hallar $\delta(r_{new}, r^*)$

 % Se halla la distancia al óptimo.

Si $\delta(r_{new}, r^*) = nc$.

 % El *tour* ya tiene distancia al óptimo buscada (nc).

 Salida = 1.

Fin si.

Fin mientras.

4 Análisis de la estructura del espacio de soluciones del *TSP*

4.1 Problemas estudiados

Se utilizaron un total de 20 problemas con soluciones óptimas ya conocidas. Estos problemas ampliamente difundidos [6], se detallan a continuación en la Tabla 1 que muestra el nombre del problema, el número n de ciudades, los parámetros NT y NT' , así como la cantidad de *tours* analizados.

La cantidad $NT' = 24$ fue elegida para permitir obtener una cantidad apreciable de soluciones en los problemas pequeños. En los problemas de mayor tamaño se disminuyó NT' debido al enorme tiempo de procesamiento que se requiere para poblaciones muy grandes.

Por falta de suficientes recursos computacionales, no se realizó la búsqueda de soluciones con $3\text{-}opt$ en los problemas de más de 101 ciudades, lo que no invalida las conclusiones del trabajo, dada la uniformidad de los resultados experimentales que serán mostrados en la siguiente sección.

Nombre del problema en TSPLIB	Número de ciudades del problema (n).	Tours en cada subconjunto (NT') (usando $Dist$)	Tamaño de la Población P (usando $Dist$)	Tamaño de la Población P (NT) (usando $3\text{-}opt$)
<i>fri26</i>	26	24	600	1000
<i>bayg29</i>	29	24	672	1000
<i>bays29</i>	29	24	672	1000
<i>att48</i>	48	24	1128	1000
<i>eil51</i>	51	24	1200	1000
<i>berlin52</i>	52	24	1224	1000
<i>st70</i>	70	24	1656	1000
<i>pr76</i>	76	24	1800	1000
<i>eil76</i>	76	24	1800	1000
<i>kroa100</i>	100	24	2376	1000
<i>kroc100</i>	100	24	2376	1000
<i>krod100</i>	100	24	2376	1000
<i>eil101</i>	101	24	2400	1000
<i>lin105</i>	105	24	2496	
<i>ch130</i>	130	24	3096	
<i>ch150</i>	150	24	3576	
<i>tsp225</i>	225	24	5376	
<i>a280</i>	280	24	6696	
<i>pcb442</i>	442	9	3969	
<i>pr1002</i>	1002	6	6006	

Tabla 1: Resumen de problemas estudiados.

4.2 Presentación de los resultados

En las poblaciones halladas con el algoritmo $Dist$ no se consideraron individuos repetidos, es decir, las poblaciones solo contenían una vez a cada solución encontrada. En cambio, en las poblaciones halladas con la heurística $3\text{-}opt$ se hallaron 1000 individuos diferentes en la mayoría de los problemas, excepto en los siguientes casos:

- En *fri26*, donde se obtuvieron 687 individuos diferentes y se alcanzó el óptimo en 87 ocasiones.
- En *bayg29*, donde se encontraron 877 individuos distintos y se llegó al óptimo 57 veces, y
- en *bays29*, donde se obtuvieron 897 tours no repetidos y se encontró el óptimo en 35 ocasiones.

Cabe destacar que en ninguno de los demás problemas considerados se halló el óptimo. Para cada población P se calcularon los siguientes valores de correlación, mostrados en la Tabla 2:

- La correlación entre la distancia al óptimo $\delta(r, r^*)$ y la longitud del tour $l(r)$, denotado como $\rho(\delta(r, r^*); l(r))$.
- La correlación entre la distancia media a la población $\delta(P, r)$ y la longitud del tour $l(r)$ denotado como $\rho(\delta(P, r); l(r))$.
- La correlación entre la distancia al óptimo $\delta(r, r^*)$ y la distancia media a la población $\delta(P, r)$ denotado como $\rho(\delta(r, r^*); \delta(P, r))$.

Como puede observarse en la Tabla 2, las correlaciones calculadas son muy significativas en todos los casos, siendo en general más altas para las poblaciones uniformes generadas con el algoritmo $Dist$ que para las poblaciones de óptimos locales generados con la heurística $3\text{-}opt$. Cabe notar además que el valor de estas correlaciones crece en general con el tamaño n del problema, por lo que se puede presumir que en general, estas correlaciones aumentan

con la complejidad del problema. Por lo que resulta razonable intentar analizar estos resultados experimentales para intentar conocer la estructura del *TSP*, especialmente para el caso de problemas de creciente complejidad. El valor t estadístico dado en la Tabla 2 indica si las correlaciones halladas son estadísticamente significativas [7]. Un valor de t mayor a 2 representa un nivel de confianza del 95 % de la correlación hallada, un valor mayor a 2,6 indica un nivel del 99 % de confianza [7]. De los elevados valores de t calculados se concluye que las correlaciones halladas para los conjuntos de soluciones generadas son altamente significativas en términos estadísticos.

Antes de iniciar el referido estudio se presentan las figuras 1 al 8, ilustrando para algunos de los problemas tipo estudiados las características resaltantes de las 3 variables estudiadas. Debido a la gran cantidad de problemas estudiados, resulta imposible presentar figuras para cada problema analizado. Seguidamente se presentan los gráficos de los problemas *fri26*, *berlin52*, *kroa100*, *pcb442* y *pr1002*.

Para cada problema se muestran tres gráficos de las variables estudiadas. Las figuras 1, 2 y 3 presentan los resultados usando la heurística *3-opt* y las figuras 4 al 8 los resultados usando el algoritmo *Dist*.

Correlaciones experimentales para poblaciones halladas con el algoritmo <i>Dist</i>						
	$\rho(\delta(r, r^*); l(r))$	t-estadístico	$\rho(\delta(P, r); l(r))$	t-estadístico	$\rho(\delta(r, r^*); \delta(P, r))$	t-estadístico
<i>fri26</i>	0,954481	78,25	0,956200	79,88	0,9996899	981,69
<i>bayg29</i>	0,961339	90,37	0,960098	88,86	0,9997985	1289,29
<i>bays29</i>	0,960703	89,59	0,958035	86,51	0,9998415	1453,57
<i>att48</i>	0,971702	138,04	0,971641	137,88	0,9999622	3857,84
<i>eil51</i>	0,979951	170,24	0,979936	170,18	0,9999725	4670,22
<i>berlin52</i>	0,978348	165,25	0,978173	164,56	0,9999643	4138,62
<i>st70</i>	0,986453	244,56	0,986430	244,35	0,9999886	8500,82
<i>pr76</i>	0,986787	258,25	0,986804	258,42	0,9999920	10593,00
<i>eil76</i>	0,987030	260,71	0,987038	260,79	0,9999913	10177,17
<i>kroa100</i>	0,988073	312,64	0,988061	312,48	0,9999961	17517,70
<i>kroc100</i>	0,987990	311,55	0,987992	311,57	0,9999950	15348,25
<i>krod100</i>	0,988899	324,26	0,988868	323,81	0,9999960	17246,37
<i>eil101</i>	0,990129	345,94	0,990111	345,62	0,9999962	17841,10
<i>lin105</i>	0,988243	322,79	0,988230	322,62	0,9999968	19602,05
<i>ch130</i>	0,992741	459,13	0,992753	459,51	0,9999975	25086,48
<i>ch150</i>	0,994288	556,91	0,994288	556,92	0,9999985	34234,79
<i>tsp225</i>	0,995618	780,47	0,995620	780,65	0,9999993	62570,48
<i>a280</i>	0,996457	969,35	0,996458	969,50	0,9999995	85598,58
<i>pcb442</i>	0,998166	1038,53	0,998165	1038,30	0,9999996	69147,23
<i>pr1002</i>	0,999130	1856,80	0,999130	1856,77	0,9999999	165631,37
Correlaciones experimentales para poblaciones halladas con heurística <i>3-opt</i>						
	$\rho(\delta(r, r^*); l(r))$	t-estadístico	$\rho(\delta(P, r); l(r))$	t-estadístico	$\rho(\delta(r, r^*); \delta(P, r))$	t-estadístico
<i>fri26</i>	0,797405	41,75	0,823567	45,87	0,786186	40,19
<i>bayg29</i>	0,757378	36,64	0,874653	57,00	0,885846	60,32
<i>bays29</i>	0,696563	30,67	0,867722	55,15	0,778338	39,16
<i>att48</i>	0,594753	23,37	0,820724	45,38	0,750131	35,84
<i>eil51</i>	0,657251	27,55	0,915290	71,79	0,753567	36,21
<i>berlin52</i>	0,715194	32,33	0,864923	54,44	0,849118	50,78
<i>st70</i>	0,607177	24,14	0,899532	65,05	0,678464	29,18
<i>pr76</i>	0,708200	31,69	0,880775	58,76	0,833839	47,72
<i>eil76</i>	0,695594	30,59	0,941236	88,04	0,772630	38,45
<i>kroa100</i>	0,691580	30,25	0,903899	66,76	0,848029	50,55
<i>kroc100</i>	0,699575	30,93	0,923682	76,16	0,795789	41,51
<i>krod100</i>	0,698733	30,86	0,912419	70,43	0,792240	41,02
<i>eil101</i>	0,707880	31,66	0,944461	90,79	0,773584	38,57

Tabla 2: Resumen de los resultados experimentales.

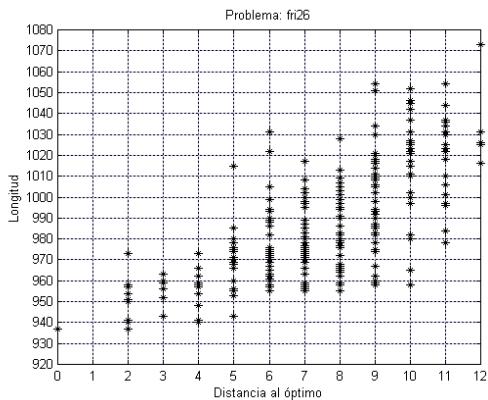


Figura 1.1: $l(r)$ en función de $\delta(r, r^*)$

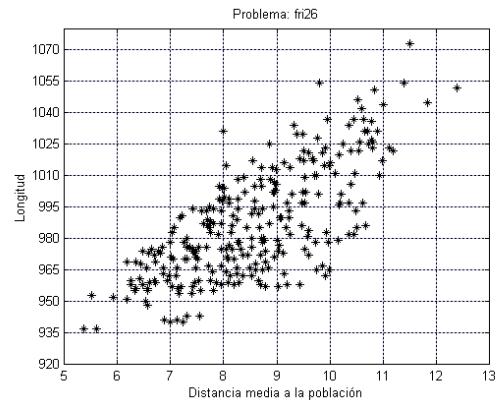


Figura 1.2: $l(r)$ en función de $\delta(P, r^*)$

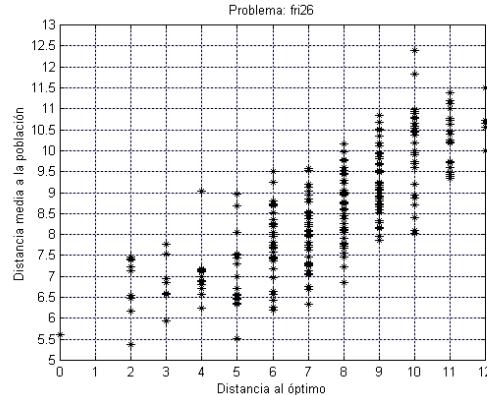


Figura 1.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 1: Problema fri26 usando 3-opt

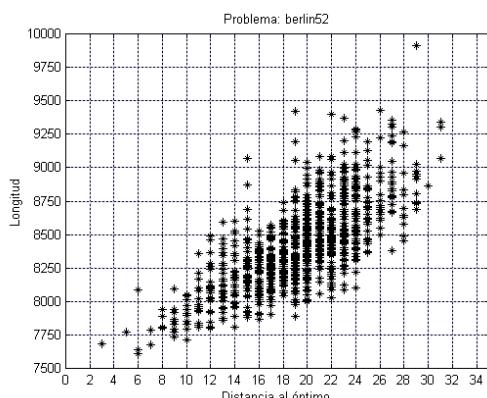


Figura 2.1: $l(r)$ en función de $\delta(r, r^*)$

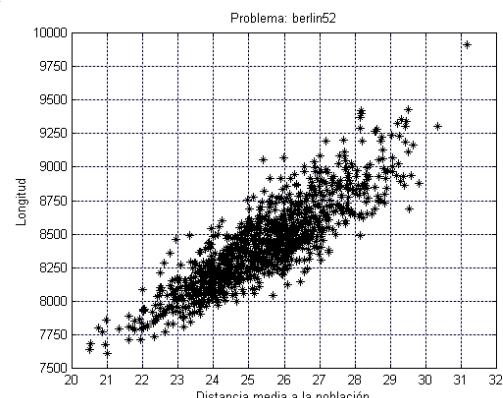


Figura 2.2: $l(r)$ en función de $\delta(P, r^*)$

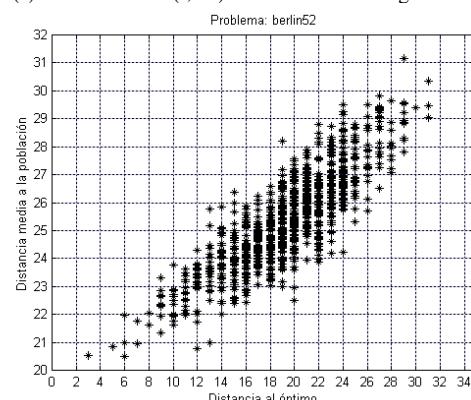


Figura 2.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 2: Problema berlín52 usando 3-opt

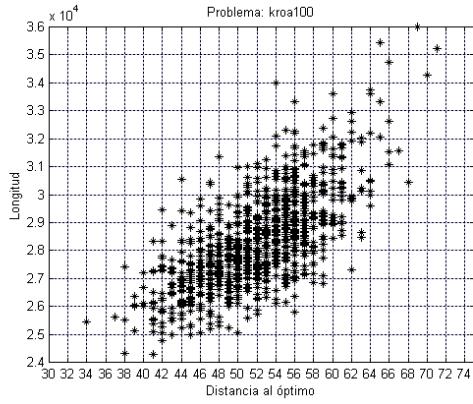


Figura 3.1: $l(r)$ en función de $\delta(r, r^*)$

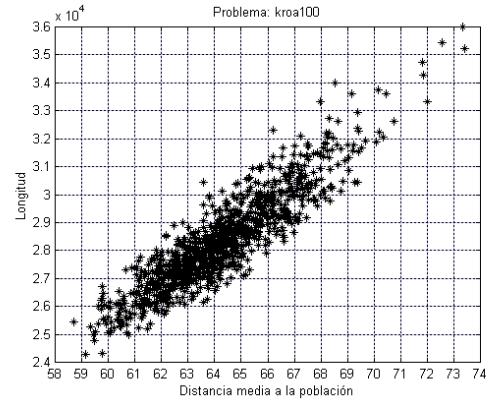


Figura 3.2: $l(r)$ en función de $\delta(P, r^*)$

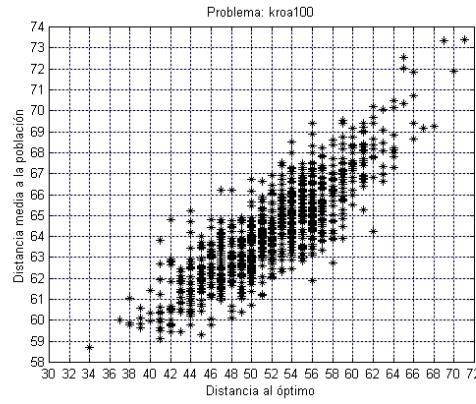


Figura 3.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 3: Problema kroa100 usando 3-opt

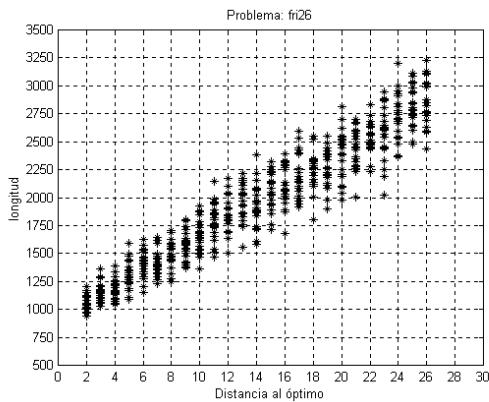


Figura 4.1: $l(r)$ en función de $\delta(r, r^*)$

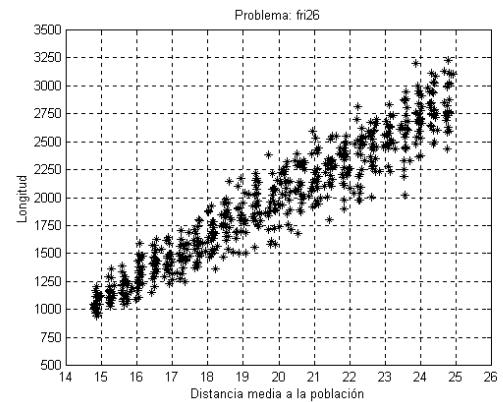


Figura 4.2: $l(r)$ en función de $\delta(P, r^*)$

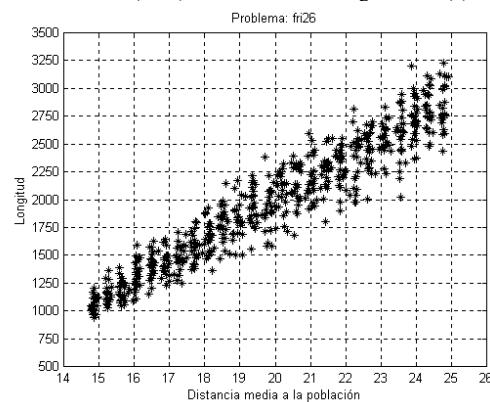


Figura 4.3 $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 4: Problema fr126 usando Dist

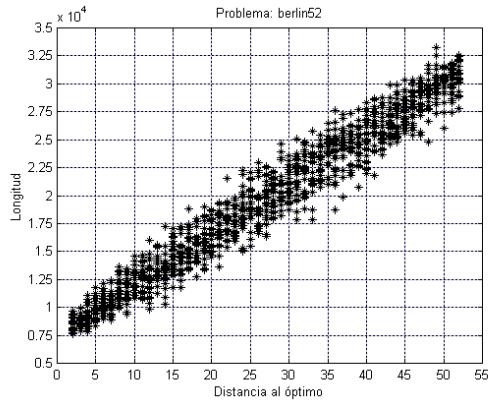


Figura 5.1: $l(r)$ en función de $\delta(r, r^*)$

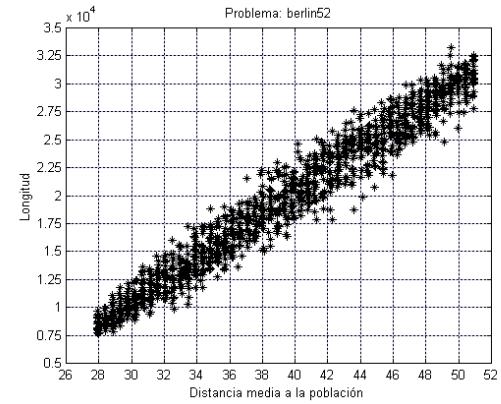


Figura 5.2 $l(r)$ en función de $\delta(P, r^*)$

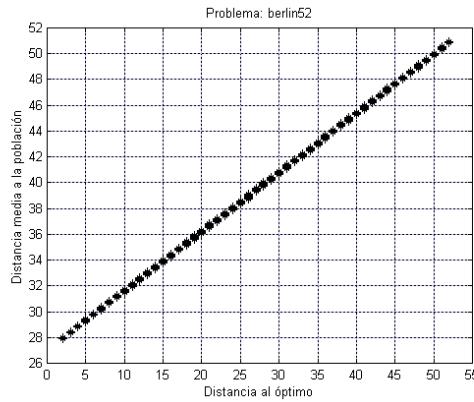


Figura 5.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 5: Problema *berlin52* usando *Dist*

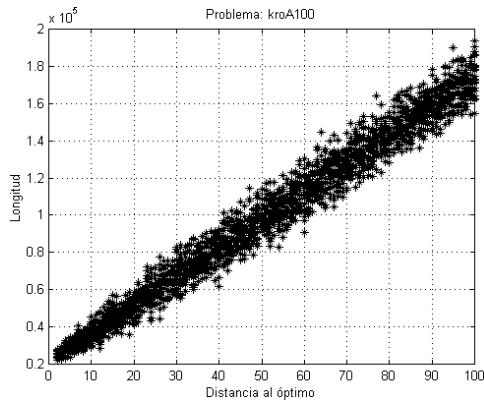


Figura 6.1: $l(r)$ en función de $\delta(r, r^*)$

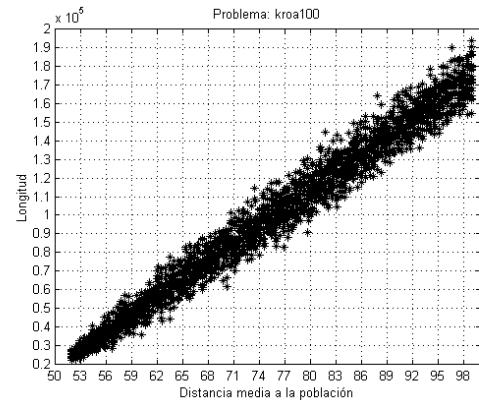


Figura 6.2: $l(r)$ en función de $\delta(P, r^*)$

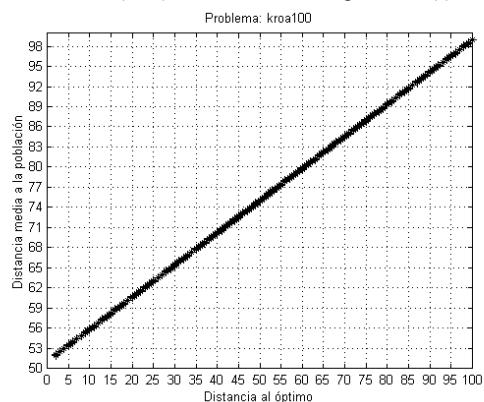


Figura 6.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 6: Problema *kroA100* usando *Dist*

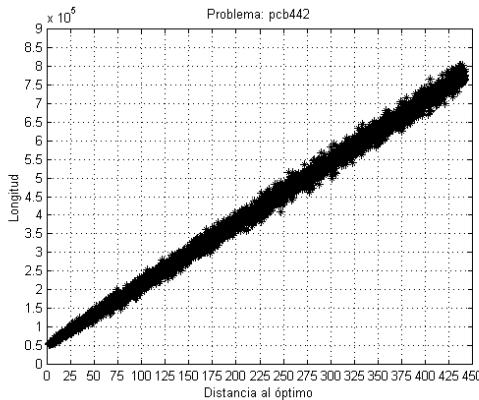


Figura 7.1: $l(r)$ en función de $\delta(r, r^*)$

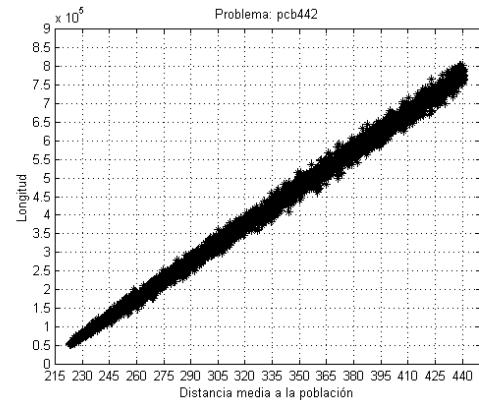


Figura 7.2: $l(r)$ en función de $\delta(P, r^*)$

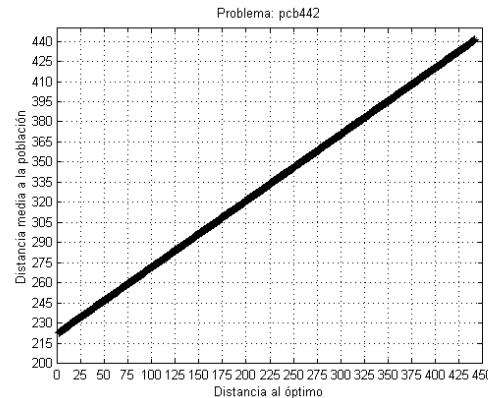


Figura 7.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 7: Problema pcb442 usando *Dist*

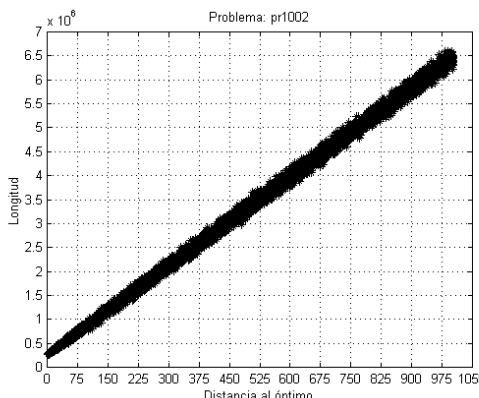


Figura 8.1: $l(r)$ en función de $\delta(r, r^*)$

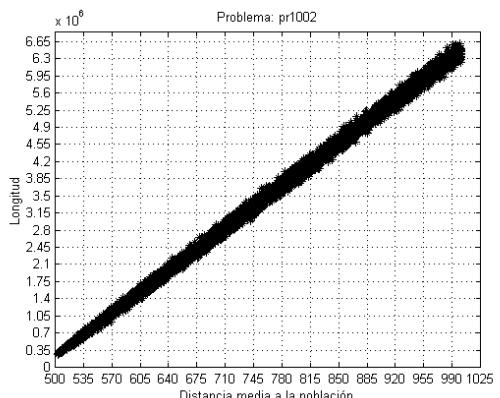


Figura 8.2: $l(r)$ en función de $\delta(P, r^*)$

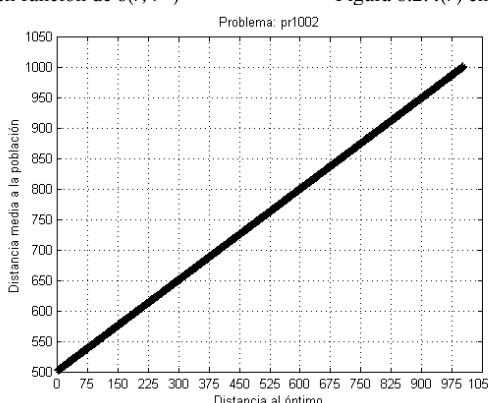


Figura 8.3: $\delta(P, r^*)$ en función de $\delta(r, r^*)$

Figura 8: Problema pr1002 usando *Dist*

5 Definición de espacio globalmente convexo

En una investigación previa, Boese [1] sugiere una interpretación intuitiva de un espacio globalmente convexo, haciendo una analogía con una estructura de *gran valle* en un contexto de minimización. Esto es, con un solo mínimo perceptible desde lejos, pero con picos y valles (mínimos locales) visto desde cerca. Una representación gráfica simplificada de esta interpretación puede observarse en la figura 9.

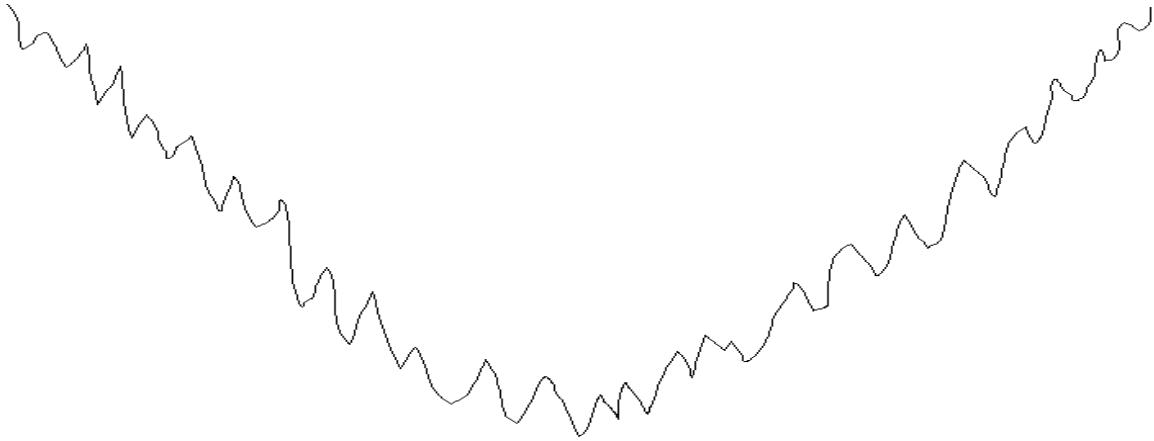


Figura 9: Representación gráfica de un espacio globalmente convexo.

Como puede notarse en la figura 9, si tomamos una población uniformemente distribuida en el dominio de dicha función globalmente convexa, o entre los óptimos locales de ésta, esta población tendría exactamente las mismas características que las encontradas en todos los problemas del TSP estudiados, es decir, se obtendrían valores muy significativos de las 3 correlaciones estudiadas y ampliamente graficadas en el presente trabajo:

1. $\rho(\delta(r, r^*); l(r))$ pues claramente las longitudes decrecen al acercarse a la solución óptima r^* .
2. $\rho(\delta(P, r); l(r))$ pues las mejores soluciones están en el centro del valle y las peores soluciones están en la periferia, conforme fuera explicado al final de la sección 2.
3. $\rho(\delta(r, r^*); \delta(P, r))$ como lógica consecuencia de las altas correlaciones anteriores, indicando además que el óptimo r^* estaría en una región central del espacio de búsqueda, rodeado de muy buenas soluciones.

Otra característica de un espacio globalmente convexo, que puede inferirse de la figura 9, es que si tomamos muestras de los óptimos locales de la zona central del mismo, las correlaciones entre las variables estudiadas serán menores que para una población de muestras uniformemente distribuidas en todo el dominio.

Como se muestra en las figuras 1 al 3, el *3-opt*, aun al comenzar en puntos totalmente aleatorios, terminó hallando soluciones concentradas en una zona relativamente cercana al óptimo para todas las instancias estudiadas. Ésta podría ser la razón por la cual las correlaciones para los individuos hallados con *3-opt* son un poco menores que las calculadas para los individuos hallados con *Dist*. Además cabe resaltar que todas las correlaciones halladas son altamente significativas estadísticamente hablando.

En consecuencia, el presente trabajo presenta suficientes datos experimentales como para confirmar que la estructura del espacio de soluciones del *TSP* tiene en general una estructura globalmente convexa, conforme a lo sugerido sin demostración por Boese [1,2]. Lógicamente, no se pudo afirmar que esta estructura globalmente convexa se da necesariamente en todos los casos, pero definitivamente, los resultados experimentales demuestran que sí se dan en la mayoría de los casos, y de hecho, los autores no lograron encontrar siquiera una instancia del *TSP* que no sea globalmente convexa.

6 Conclusiones

Los resultados muestran unas muy altas correlaciones entre la $l(r)$ y su $\delta(r, r^*)$ en todas las poblaciones halladas con *Dist*. Esto claramente indica que a medida que un *tour* comparte un mayor número de arcos con el óptimo su longitud tiende a disminuir. Dicho de otra forma la posibilidad de hallar una buena solución es mayor mientras más cerca del óptimo r^* se busque. El estudio de la correlación entre $\delta(P, r)$ y $l(r)$ refuerza la idea de que las soluciones buenas se encuentran cerca unas de otras y en una zona central del espacio solución.

Esta propiedad resulta de especial importancia al intentar explicar el excelente desempeño de los algoritmos evolutivos basados en poblaciones, como las colonias de hormigas y los algoritmos genéticos, que al trabajar con una población que va optimizando el valor de $l(r)$, de acuerdo a lo aquí expuesto estaría concentrando su búsqueda

en la zona central del espacio de búsqueda, donde en efecto se encuentran las mejores soluciones, y r^* en particular, dada la característica globalmente convexa del espacio de soluciones [3].

En futuros trabajos se intentará dar una definición más formal de un espacio globalmente convexo o en su defecto definir grados de convexidad global para una función dada.

En resumen, el presente trabajo aporta por primera vez, amplios resultados experimentales que soportan la conjectura de un espacio de búsqueda globalmente convexo, en el caso del TSP, lo que explicaría el excelente desempeño de diversos algoritmos basados en poblaciones, a la hora de resolver el Problema del Cajero Viajante.

Referencias

- [1] Boese, K.D.: Cost versus Distance in the Traveling Salesman Problem. Reporte Técnico 950018, University of California, Los Angeles, Computer Science Department, (1995).
- [2] Boese, K.D, Kahng, A.B. y Muddu, S.: A new Adaptive Multi-Start Technique for Combinatorial Global Optimization, Operations Research Center. Vol 16, pp. 101-113, (1994).
- [3] Gómez O.y Barán B.. Reasons of ACO's Success in TSP. En M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, y T. Stutzle, editores, Proceedings of ANTS 2004 - Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence. A ser publicado, tomo 3172 de LNCS. Springer-Verlag, Brussels, septiembre 2004.
- [4] Johnson D.S. y McGeoch L.A.: The traveling salesman problem: A case of study in local optimization, Local Search in Comb. Optimization, Eds. New York: Wiley: New York, (1997).
- [5] Stützle, T. y Hoos, H.H.: Max-Min Ant System. Future Generation Computer Systems. Vol 16, pp. 889-914 , (2000).
- [6] TSPLIB: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- [7] Walpole R. Y Myers R.: Probabilidad y estadística. McGraw-Hill/Interamericana de México, S.A. de C. V (1991).