

# *ASrank*: un análisis de robustez

Sergio Fernando Sanz Troianai Gabriela F. Minetti

Universidad Nacional de La Pampa (Argentina),  
Facultad de Ingeniería,  
Laboratorio de Investigación en Sistemas Inteligentes  
e-mail: fstanando@gmail.com, minettig@ing.unlpam.edu.ar

**Resumen** La resolución de los problemas de optimización en el mundo real es dificultosa no sólo por lo que significa la búsqueda de soluciones en espacios de grandes dimensiones sino, también, por el alto grado de incertidumbre que afecta a los datos de entrada al problema. Para ello son necesarios algoritmos que encuentren soluciones de muy buena calidad en poco tiempo y que además sean robustos. Los algoritmos metaheurísticos han demostrado una y otra vez que son capaces de lograr el primer objetivo, pero no existen muchas investigaciones que estudien y analicen la robustez de los mismos. En este trabajo se analiza la resistencia al ruido (o robustez) de una variante de los algoritmos de optimización basados en colonias de hormigas, denominado *AS<sub>rank</sub>*. Para ello se utiliza el problema del viajante de comercio, ya que modela un amplio espectro de problemas del mundo real. También se propone *genTSP*, un generador de instancias ruidosas del mencionado problema. El análisis de los resultados obtenidos en la experimentación arroja que el algoritmo estudiado es robusto y no necesita esfuerzo computacional extra para alcanzar este segundo objetivo.

**Keywords:** Optimización basada en colonia de hormigas, *ASrank*, TSP, robustez

## 1. Introducción

En los problemas de optimización del mundo real es común encontrar un alto grado de incertidumbre (ruido), debido a la complejidad de la generación de soluciones candidatas, a los errores en los procesos de medición, a los problemas con entornos con un alto grado de dinamismo, a las funciones objetivo ruidosas, entre otros motivos [9]. El efecto del ruido sobre el desempeño de las técnicas de optimización es un área de investigación aún poco estudiada pero de mucha utilidad en diversos campos de aplicación. En el caso de las técnicas metaheurísticas se pueden encontrar estudios sobre: optimización de simulación de sistemas estocásticos complejos [7], problemas del camino más corto estocástico [8], ensamblado de fragmentos de ADN ruidosos [11], problema oneMax [6] y detección de bordes en imágenes con ruido [1].

En particular, los algoritmos de optimización metaheurísticos basados en colonias de hormigas (*Ant Colony Optimization*, ACO) muestran una buena adaptación a los problemas de incertidumbre, ruido y dinamismo debido a su naturaleza dinámica y distribuida [3]. El desarrollo de este trabajo de investigación consiste en analizar la resistencia al ruido, también conocida como robustez, de una variante del algoritmo metaheurístico basado en la optimización de colonia de hormigas, denominado *AS<sub>rank</sub>* [2]. Para llevar a cabo dicho análisis se trabajará con una de las fuentes de ruido que se presenta con mayor frecuencia en los problemas de optimización del mundo real y que es más difícil de detectar. Esta fuente de ruido surge por los errores cometidos en los procesos de medición, por los cuales se obtienen los datos de entrada a cualquier algoritmo de optimización. Consecuentemente,

se resolverá el problema del viajante de comercio (*Traveling Salesman Problem*, TSP) [13], ya que modela diferentes situaciones reales (planificación de tareas, ruteo de redes de telecomunicaciones, diseño de rutas para la recolección de residuos, diseño de circuitos impresos, etc.) y es de difícil resolución dado que pertenece al grupo de problemas NP-duros.

Para verificar si  $AS_{rank}$  es resistente al ruido son necesarias instancias del problema que presenten esta característica. Pero, las instancias de TSP publicadas en la literatura no contemplan ruido en sus datos, por ende en este trabajo también se propone un generador de instancias de TSP que simulen errores en la medición de las distancias. Por consiguiente, para estudiar la robustez de  $AS_{rank}$  se analizan las diferencias entre las soluciones encontradas para las instancias sin y con ruido. Si no se detectan diferencias (estadísticamente significativas), el algoritmo muestra un comportamiento neutro (insensible, indistinto) a pequeñas variaciones en los datos de entrada. Consecuentemente, este algoritmo se considera robusto para resolver instancias ruidosas.

El artículo se organiza de la siguiente manera. En la sección 2 se introducen los conceptos generales de  $AS_{rank}$  y se explica la versión aquí propuesta. En tanto que, en la sección 3 se describe el problema usado en la experimentación y el generador de instancias ruidosas formulado en este trabajo. En el siguiente apartado se especifica el diseño experimental, el ambiente de ejecución y la metodología de análisis utilizados, para llevar a cabo la experimentación y el análisis de los resultados. En la sección 5 se presentan los resultados obtenidos y el análisis de los mismos. En la última sección se resumen las conclusiones y se proyectan trabajos futuros.

## 2. Sistema de hormigas basado en jerarquización (*Rank-based Ant System*, $AS_{rank}$ )

El sistema de hormigas basado en jerarquización propuesto por Bullenheimer et al. en [2] es una extensión del sistema de hormigas desarrollado por Dorigo et al. en [4]. Para una mejor comprensión de  $AS_{rank}$ , primero se presenta una descripción de  $AS$  y luego se explica dicha extensión.

$AS$  se caracteriza por el hecho que la actualización de la feromona se realiza una vez que todas las hormigas completaron sus soluciones. Primero, todos los caminos de feromonas son reducidos por un factor constante, implementando de esta manera la evaporación de feromona. Segundo, toda hormiga de la colonia deposita una cantidad de feromona en función de la calidad de su solución. Inicialmente  $AS$  no usa acciones de demonio, pero es aconsejable hacerlo al agregar una búsqueda local para refinar las soluciones generadas por las hormigas.

Las soluciones en  $AS$  son construidas de la siguiente manera. A cada paso de la construcción, una hormiga  $k$  en  $AS$  elige ir del nodo  $r$  al  $s$  con una probabilidad que es calculada como:

$$p_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in N_r^k} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{si } s \in N^k(r) \\ 0, & \text{otro caso} \end{cases}$$

Donde  $N_k(r)$  es el vecino factible de la hormiga  $k$  cuando está en el nodo  $r$ ,  $\tau_{rs}$  es la cantidad de feromona perteneciente a la conexión  $rs$ ,  $\eta_{rs}$  es uno sobre la distancia entre  $r$  y  $s$ , y  $\alpha$  y  $\beta \in \mathbb{R}$  son dos parámetros que pesan la importancia relativa del camino de feromona y la información heurística, respectivamente.

El depósito de feromona es realizado una vez que todas las hormigas han finalizado de construir sus soluciones. Primero, el camino de feromona asociado a cada arco es evaporado reduciendo todas las feromonas por un factor constante, conocido como tasa de evaporación, ( $\rho \in (0, 1]$ ):

$$\tau_{rs} \leftarrow (1 - \rho) \cdot \tau_{rs}$$

Luego, cada hormiga remarca el camino que ha seguido y deposita una cantidad de feromona por cada conexión recorrida:

$$\tau_{rs} \leftarrow \tau_{rs} + \Delta\tau_{rs}^k, \quad \forall a_{rs} \in S_k$$

Donde  $\Delta\tau_{rs}^k = f(C(S_k))$ , es la cantidad de feromona depositada que depende de la calidad de la solución,  $C(S_k)$ , construída por la hormiga  $k$ .

En  $AS_{rank}$ , cada hormiga deposita una cantidad de feromona que disminuye con su rango (o jerarquía), siendo la mejor hormiga de cada iteración la que deposita la mayor cantidad de feromona. Esta actualización se realiza globalmente como sigue:

1. Se seleccionan las  $w$  hormigas para la jerarquización de acuerdo a la calidad de sus soluciones:  $(S'_1, \dots, S'_w)$ , siendo  $S'_1$  la mejor solución generada hasta el momento.
2. Se deposita feromona en las conexiones recorridas por las mejores hormigas (denominadas hormigas elitistas). La cantidad de feromona depositada depende directamente de la clasificación de la hormiga y de la calidad de la solución. Es decir, que la feromona depositada por la hormiga clasificada en la posición  $r$  se multiplica por  $(w - r)$ .
3. Las conexiones recorridas por la mejor solución global reciben una cantidad adicional de feromona que depende de la calidad de la solución. Este depósito de feromona es considerado el más importante, por lo tanto se multiplica por  $w$ .

En la versión de  $AS_{rank}$  implementada en este trabajo (ver algoritmo 1), primero se inicializan los parámetros y el rastro de feromona, en un segundo paso, cada hormiga construye, iterativamente, una solución al problema. A continuación,  $AS_{rank}$ , aplica el algoritmo de búsqueda local 3 – *opt* a cada una de estas soluciones. Finalmente, aplica los tres pasos que caracterizan a  $AS_{rank}$  (descriptos en el párrafo anterior) y vuelve al paso dos hasta que el número máximo de iteraciones se haya ejecutado.

---

#### Algoritmo 1 $AS_{rank}$

---

```

Inicializar parámetros y rastro de feromonas ( $\tau$ );
 $t = 0$ ;
repeat
  for  $i = 0$  to  $\mu$  do
    { $\mu$  es la cantidad de hormigas}
    La hormiga  $i$  construye paso a paso una solución;
    Aplicar 3 – opt a cada solución;
  end for
  Elegir las  $w$  mejores soluciones y ordenarlas de forma decreciente  $(S'_1, \dots, S'_w)$ ;
  Actualizar el rastro de feromonas:
  1) Depositar feromona en las conexiones recorridas por cada  $S'_r$  en función a su calidad y a su posición;
  2) Depositar feromona en las conexiones recorridas por la mejor solución global en función a su calidad;
   $t = t + 1$ ;
until  $t = K$ 
return  $S'_1$ ;

```

---

### 3. Generación de instancias con ruido

Como se dijo anteriormente, para determinar si  $AS_{rank}$  es robusto frente a errores en los datos de entrada del problema se requieren instancias de TSP que presenten dicha característica. Dado que las instancias de TSP publicadas en la literatura no introducen errores en sus datos, se propone un generador de instancias de TSP, denominado *genTSP*, que simule los errores en la medición de las distancias. Consecuentemente, en esta sección se introduce *genTSP* y las nuevas instancias generadas, siendo necesario introducir previamente la definición del problema del viajante de comercio, las instancias sin ruido publicadas en la literatura y la descripción de los errores que se cometen en la medición de distancias.

El problema del viajante de comercio puede ser descrito como: dado un conjunto de ciudades  $C_1, C_2, C_3, \dots, C_n$  y un conjunto de distancias  $d(C_i, C_j)$  para cada par de ciudades distintas  $(C_i, C_j)$ , consiste en encontrar un recorrido de distancia mínima que pase sólo una vez por cada una de las ciudades y, finalmente, regrese a la ciudad de la que partió. A fin de obtener un conjunto de prueba de significativa complejidad, en este trabajo se seleccionaron instancias de TSP de variadas características y con gran cantidad de ciudades.

Las instancias en cuestión se muestran en la tabla 1 y pueden descargarse de la librería TSPLIB<sup>1</sup>. En dicha tabla se especifica el nombre, la dimensión y el valor óptimo para cada una de las instancias. Es importante aclarar que estas instancias serán identificadas con la etiqueta “0%”, ya que son los casos sin ruido.

<i>Nombre</i>	<i>Dimensión</i>	<i>Valor óptimo</i>
d198	198	15780
a280	280	2579
lin318	318	42029
pcb442	442	50778
rat783	783	8806
pcb1173	1173	56892
d1291	1291	50801
fl1577	1577	22249

**Cuadro 1.** Instancias seleccionadas de TSP de la librería TSPLIB.

Los errores en la medición surgen dependiendo del método o instrumento utilizado para determinar la distancia entre las ciudades. Existen diversos instrumentos para medir distancias [10], pero para obtener la longitud de las rutas se utilizan generalmente odómetros mecánicos o eléctricos y, también, sistemas de información geográfica (SIGs). En el proceso de medición se introducen errores ya sea por una mala calibración de los instrumentos o los denominados errores sistemáticos y aleatorios que son inevitables. Los errores sistemáticos están relacionados con la pendiente, la graduación, la temperatura, la tensión y la catenaria. En tanto que, los errores aleatorios se vinculan con la alineación y la verticalidad. Además están los errores llamados groseros, que pueden cometerse por equivocación del operador y que son impredecibles. Entre los más comunes se pueden

<sup>1</sup> <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp/>

encontrar: la identificación incorrecta de un punto, la lectura y/o anotación errónea por transposición de números y los cálculos aritméticos mentales inexactos. Por último es necesario considerar los errores accidentales ya que son totalmente inevitables y sus causas no son conocidas.

En virtud de todos los tipos de errores presentes en la medición de distancias, en este trabajo se consideran cuatro niveles de ruido en las instancias de TSP. De esta manera, *genTSP* crea 4 nuevos grupos de instancias a partir de los casos sin ruido e incorpora a cada una de las distancias errores de hasta un 5, 10, 15 y 20%. Estos nuevos cuatro grupos de instancias son identificados con las etiquetas “5%”, “10%”, “15%” y “20%”, respectivamente. El algoritmo 2 muestra un pseudocódigo del generador *genTSP*. Este utiliza la distribución uniforme para crear en forma aleatoria un error  $\epsilon \in [-l, l]$  para cada distancia entre dos ciudades. Donde  $l = \gamma \times d(C_i, C_j) \forall i \neq j, i, j \leq n$  y  $\gamma \in \{5\%, 10\%, 15\%, 20\%\}$ , los niveles de ruido.

---

**Algoritmo 2** *genTSP*

---

```

for  $i = 0$  to  $n$  do
  for  $j = 0$  to  $n$  do
    if  $i \neq j$  then
       $l = \gamma \times d(C_i, C_j)$ ;
       $\epsilon = \text{uniform.random}(-l, l)$ ;
       $d(C_i, C_j) = d(C_i, C_j) + \epsilon$ ;
    end if
  end for
end for
return

```

---

## 4. Metodología de experimentación

En esta sección se describe el diseño experimental usado en este trabajo, el ambiente de ejecución y la metodología de análisis. La variante de  $AS_{rank}$  propuesta se evaluó con el grupo de las instancias sin ruido y los cuatro grupos de instancias con ruido, sumando un total de 40 instancias ( $8 \text{ instancias} \times 5 \text{ grupos}$ ). Además al considerarse la naturaleza estocástica de los algoritmos metaheurísticos, cada instancia fue resuelta 30 veces por esta versión de  $AS_{rank}$ .

Los valores paramétricos,  $\alpha = 1$  y  $\beta = 2$  utilizados para evaluar el rendimiento de esta metaheurística son determinados según lo publicado en [5]. En tanto que, los parámetros  $\rho = 0,2$ ,  $w = 8$ , el tamaño de la población de hormigas  $\mu = 25$ , el número de iteraciones realizadas por la búsqueda local,  $k = 5000$  y el número total de iteraciones llevado a cabo por  $AS_{rank}$ ,  $K = 10000$  surgieron de la combinación de diferentes valores que arrojó los mejores resultados en una experimentación previa. Durante la misma, se ejecutó  $AS_{rank}$  bajo todas las configuraciones que surgen al combinar los diferentes valores propuestos para cada uno de estos parámetros. El conjunto de valores considerados son: para  $\rho = \{0,1,0,2,0,3\}$ , para  $\sigma = \{7,8,9\}$ , para  $\mu = \{25,100\}$ , para  $k = \{0,1000,5000\}$  y, finalmente, para  $K = \{1000,10000\}$ . En tanto que, el conjunto de prueba está formado por las instancias descritas en la sección 3.

El algoritmo fue implementado en C++. Esta experimentación se lleva a cabo sobre computadoras con procesadores AMD Phenom (64 bits) a 2.4 GHz y 2 GB de RAM, con un sistema operativo perteneciente a la distribución Slackware de Linux con una versión del kernel 2.6.27.7-smp.

Para poder comparar los resultados obtenidos para las distintas instancias se mide la desviación relativa,  $DR$ , de la distancia recorrida (costo) por las mejores soluciones encontradas por el algoritmo

con respecto al valor óptimo de cada instancia. Además de posibilitar la comparación al brindar datos normalizados, esta métrica también permite analizar la calidad de las soluciones halladas. El cálculo de los valores de  $DR$  se describe en la Ecuación 1, donde  $AlgSol$  es el costo mínimo obtenido por  $AS_{rank}$  para una instancia del problema en las 30 ejecuciones y  $OptSol$  es el costo óptimo para la instancia en cuestión.

$$DR = \frac{AlgSol - OptSol}{OptSol} \quad (1)$$

Con el objetivo de evaluar la robustez de  $AS_{rank}$ , se analizan las diferencias entre las soluciones encontradas para las instancias sin y con ruido. Si no se detectan diferencias (estadísticamente significativas), el algoritmo muestra un comportamiento neutro (insensible, indistinto) a pequeñas variaciones en los datos de entrada. Consecuentemente, este algoritmo se considera robusto para resolver instancias ruidosas. Este análisis se basa en los valores de  $DR$  obtenidos para las instancias sin y con ruido, que luego son usados para comparar la calidad de las soluciones halladas en ambos casos. Por ende, si estadísticamente se detectan soluciones de calidad similar ante instancias sin y con ruido se infiere que  $AS_{rank}$  es un algoritmo resistente al ruido.

Lo anterior se traduce en la comparación estadística de las medias de dos muestras independientes. En nuestro caso sería comparar los resultados obtenidos para instancias con 0 % de ruido con los encontrados para las instancias con 5 %, luego contrastar los resultados hallados para instancias con 0 y 10 % , con 0 y 15 % y, finalmente, con 0 y 20 % de ruido. De esta forma, si para un par de muestras el test estadístico arroja que no hay diferencias significativas entre ellas significa que el algoritmo es robusto en la resolución de instancias con el nivel de ruido en cuestión.

Dado que se comparan dos muestras se emplean t-tests o el test Mann-Whitney U, según sea el tipo de distribución de datos de tales muestras. El t-test no puede aplicarse sobre cualquier distribución de datos, puesto que en el caso de una distribución no normal de datos puede ser erróneo el resultado. En este caso, lo correcto sería aplicar el test Mann-Whitney U. Para determinar si los datos siguen una distribución normal se aplican la prueba de Shapiro-Wilks y homogeneidad de varianzas (homocedasticidad). Cabe aclarar que para todos los tests estadísticos realizados en este trabajo, el valor de  $\alpha$  se fija en 0.01, para indicar un nivel de confianza del 99 % en los resultados. Por ejemplo, si el t-test arroja un valor  $p \geq \alpha$  significa que las medias de las dos muestras son estadísticamente similares, en cambio si  $p < \alpha$  entonces se trata de muestras significativamente distintas.

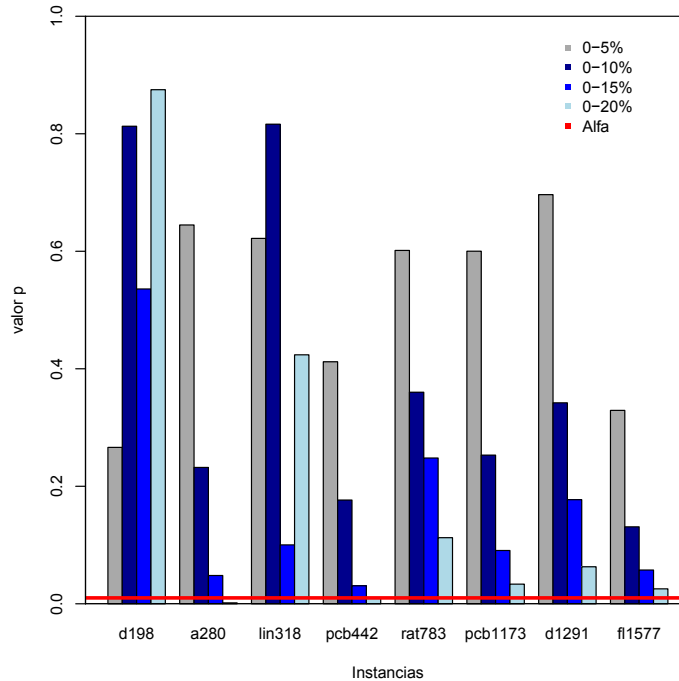
Además de estudiar la robustez, se analiza el esfuerzo numérico realizado por el algoritmo considerando el número de iteraciones que fueron necesarias para encontrar la mejor solución. Para ello se utilizan los diagramas de caja [12], ya que facilitan la comparación entre los resultados obtenidos por un algoritmo para las instancias sin y con ruido, y permiten establecer si existen o no diferencias estadísticamente significativas entre ellos.

## 5. Analisis de los resultados obtenidos

En la primera parte de esta sección se analiza la robustez de  $AS_{rank}$  por medio de los resultados arrojados por los tests estadísticos. En tanto que, en la segunda se compara el esfuerzo computacional realizado por el algoritmo al solucionar el problema con distintos niveles de ruido.

## 5.1. Análisis de robustez

La figura 1 muestra los valores  $p$  arrojados por el t-test o el test Mann-Whitney U, al comparar la desviación relativa ( $DR$ ) de cada una de las instancias sin ruido (0%) con sus pares en los diferentes niveles de ruido (5, 10, 15 y 20%, respectivamente). También indica el valor de  $\alpha = 0,01$  con propósitos comparativos mediante una recta.



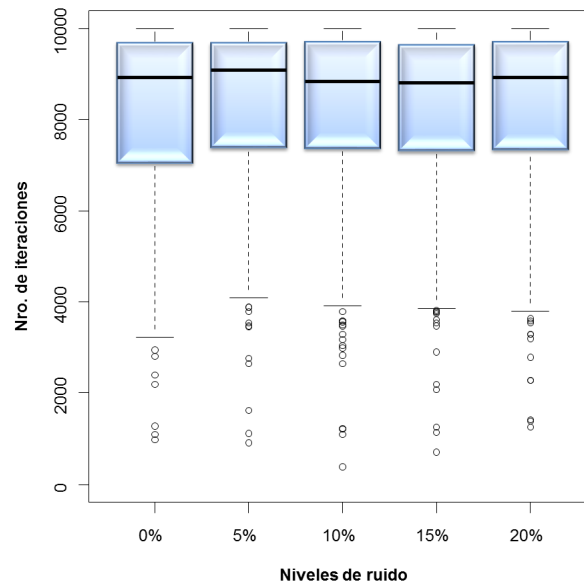
**Figura 1.** Valores  $p$  arrojados por las pruebas estadísticas al comparar la distancia relativa de las instancias con 0, 5, 10, 15 y 20% de ruido.

En primer lugar es fácilmente observable que hasta niveles del 15% de error y para todas las instancias, los valores  $p$  son todos mayores a  $\alpha$ . Esto indica que los resultados obtenidos son estadísticamente similares, comprobándose la robustez de  $AS_{rank}$  para niveles de 5, 10 y 15% de ruido. En segundo plano se puede detectar que si el nivel de error es de un 20%,  $AS_{rank}$  es robusto para el 75% de las instancias. Sólo para *a280* y *pcb442*, el algoritmo no logra ser resistente al ruido con este alto nivel de ruido.

Del análisis realizado se desprende que  $AS_{rank}$  es un método de optimización robusto, ya que la resistencia al ruido es estadísticamente corroborada en instancias del problema de diferente complejidad y con diversos niveles de ruido. Estos últimos van desde niveles de error muy bajos (5%) a muy altos (20%).

## 5.2. Comparación del esfuerzo computacional

En el apartado anterior se verificó la característica de robustez de  $AS_{rank}$ , a continuación se analiza si esta importante ventaja del algoritmo implicaría un mayor esfuerzo computacional. Para ello se compara la cantidad de iteraciones necesarias para que ASrank encuentre la mejor solución en cada nivel de ruido, considerando todas las instancias. Esta comparación se desprende de los datos representados en el diagrama de cajas de la figura 2, donde también puede observarse si las diferencias son o no estadísticamente significativas.



**Figura 2.** Diagramas de cajas (boxplots) que representan la cantidad de iteraciones utilizada por ASrank para encontrar la mejor solución frente a los diferentes niveles de ruido, comenzando por los casos sin ruido (0%).

Del análisis de la figura 2 se concluye que para todos los niveles de ruido (de 0 a 20%), el 75% de las mejores soluciones son encontradas después de las 7000 iteraciones y que, los valores promedios varían entre 8000 y 8250 iteraciones. Estos indicadores muestran fehacientemente que el esfuerzo computacional realizado por  $AS_{rank}$  para resolver los casos sin y con errores son estadísticamente similares. En consecuencia,  $AS_{rank}$  no sólo es resistente al ruido sino que además no necesita de un mayor esfuerzo numérico.

## 6. Conclusiones

En este trabajo se introduce un generador de instancias ruidosas correspondientes al TSP, con el cual se crean un conjuntos de instancias con cuatro distintos niveles de ruido. Estas nuevas instancias



son usadas para verificar la resistencia al ruido o robustez de un algoritmo metaheurístico basado en colonia de hormigas,  $AS_{rank}$ , frente a la variación en los datos de entrada.

El análisis realizado sobre los datos obtenidos en la experimentación arroja que  $AS_{rank}$  es un método de optimización robusto, ya que se corrobora estadísticamente la resistencia al ruido en instancias del problema de diferente complejidad y con diversos niveles de ruido. Estos últimos van desde niveles de error muy bajos (5 %) a muy altos (20 %). Pero,  $AS_{rank}$  no sólo es resistente al ruido sino que además no necesita de un mayor esfuerzo numérico, ya que que el esfuerzo computacional realizado para resolver los casos sin y con errores son estadísticamente similares.

Como trabajo futuro, en primer lugar se pretende utilizar los grupos de instancias generadas con ruido para analizar la robustez de otros algoritmos de optimización basados en colonia de hormigas. En segundo lugar se intentará mejorar la resolución de situaciones reales al aplicar el conocimiento adquirido en estos análisis.

## Agradecimientos

Los autores y el trabajo de investigación son financiados por la Universidad Nacional de La Pampa y la ANPCyT por medio del contrato PICTO 2011-0278 y del programa de incentivos.

## Referencias

1. Bateria, A.V., Oppus, C.: Image edge detection using ant colony optimization. WSEAS Trans. Sig. Proc. 6(2), 58–67 (Apr 2010), <http://dl.acm.org/citation.cfm?id=1853856.1853860>
2. Bullheimer, B., Hartl, R.F., Strau C.: A new rank based version of the ant system – a computational study (1997)
3. DiCaro, G.A., Ducatelle, F., Gambardella, L.M.: Reflecting Interfaces: the Complex Coevolution of Information Technology Ecosystems, chap. Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks. Idea Group, Hershey, PA, USA (2008)
4. Dorigo, M., Maniezzo, V., Coloni, A.: The ant system: Optimization by a colony of cooperating agents. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B 26(1), 29–41 (1996)
5. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Company, Scituate, MA, USA (2004)
6. Friedrich, T., Kötzing, T., Krejca, M.S., Sutton, A.M.: Robustness of ant colony optimization to noise. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 17–24. GECCO '15, ACM, New York, NY, USA (2015), <http://doi.acm.org/10.1145/2739480.2754723>
7. Gosavi, A.: The effect of noise on artificial intelligence and meta-heuristic techniques. In: of Mechanical Engineering, A.S. (ed.) Proceedings of the Artificial Neural Networks in Engineering Conference (Intelligent Engineering systems through Artificial Neural Networks). vol. 12, pp. 981–988 (2002)
8. Horoba, C., Sudholt, D.: Ant colony optimization for stochastic shortest path problems. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. pp. 1465–1472. GECCO '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1830483.1830750>
9. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. Trans. Evol. Comp 9(3), 303–317 (Jun 2005), <http://dx.doi.org/10.1109/TEVC.2005.846356>
10. Matera, L.C.: Topografía Plana. Facultad de Ingeniere la Universidad de Los Andes (2002)
11. Minetti, G., Leguizamón, G., Alba, E.: An improved trajectory-based hybrid metaheuristic applied to the noisy DNA Fragment Assembly Problem. Information Sciences 277, 273 – 283 (2014), <http://www.sciencedirect.com/science/article/pii/S0020025514001261>
12. R, M., Tukey, J.W., Larsen, W.A.: Variations of box plots. The American Statistician 32(1), 12–16 (1978)
13. Shmoys, D., Lenstra, J., Kan, A., Lawler, E.: The Traveling Salesman Problem. Wiley Interscience Series in Discrete Mathematics, John Wiley & Sons (1985)