

Un análisis preliminar de nuevos modelos de mutación dirigida en algoritmos genéticos.

B. Rodríguez-Puerta, Francisco Díaz-Barrancas, F. Chávez, F. Fernández de Vega
Dpto. de Ingeniería en Sistemas Informáticos y Telemáticos
Universidad de Extremadura
06800 Mérida, España.
Email: {brpuerta, frdiaz, fchavez, fcofdez}@unex.es

Resumen—En este artículo se presenta un análisis preliminar de un modelo de operador de mutación dirigida para problemas con codificación binaria y sin epistas. Esta versión del operador permite asociar una probabilidad de mutación a cada gen de cada individuo, proporcional a la influencia que ha tenido dicho gen sobre la calidad del individuo durante el proceso evolutivo. Estos valores de probabilidad, permiten a cada individuo realizar mutaciones de manera dirigida, con el objetivo de reducir el tiempo de convergencia del algoritmo.

El conjunto de experimentos realizado con el nuevo operador de mutación demuestra que el algoritmo genético converge a soluciones en etapas más tempranas del proceso evolutivo, en comparación con la mutación clásica. Se han llevado a cabo una serie de experimentos con un problema clásico de test, donde aplicando el nuevo operador de mutación dirigida, se consiguen buenos resultados. Aunque estos resultados son aún muy preliminares, esperamos poder continuar el estudio en problemas más complejos en el futuro, y mostrar así la utilidad de esta versión de la mutación dirigida en otros contextos.

I. INTRODUCTION

Los algoritmos evolutivos [5] (AEs), son un conjunto de técnicas de optimización y búsqueda de soluciones basados en la teoría de la evolución de Charles Darwin. Inspirados en esta teoría, utilizan una población de individuos, soluciones candidatas al problema que pretende resolverse, que tratan de adaptarse mediante evolución al entorno para resolver el problema en cuestión.

Dentro de la familia de los AEs se encuentran los algoritmos genéticos [6] (AGs). Estos nos ofrecen técnicas robustas de búsqueda y optimización para la obtención de resultados en una gran variedad de problemas. Por ejemplo, son ampliamente utilizados en los problemas conocidos como “NP completos”, donde ofrecen la posibilidad de encontrar soluciones satisfactorias en tiempos razonables. En la actualidad

los AGs son ampliamente utilizados por la comunidad científica.

Un AG típico se compone de los siguientes elementos:

- Un individuo: es la representación de una posible solución al problema.
- Una población: compuesta por un conjunto de individuos.
- Una función de fitness: que permite evaluar la calidad de un individuo.
- Operadores genéticos: que permiten generar una nueva población a partir de la anterior (operador cruce, operador mutación y operador selección).
- Parámetros de control: que permiten controlar la ejecución del algoritmo.

Para maximizar la eficiencia de los AGs es posible modificar los parámetros y operadores genéticos anteriormente mencionados.

Entre los componentes del algoritmo, hay tres parámetros fácilmente ajustables que permiten modificar el proceso de búsqueda de soluciones: probabilidad de mutación (P_m), probabilidad de cruce (P_c) y tamaño de la población (T). Son numerosos los estudios existentes que se centran en proponer heurísticas para optimizar dichos parámetros con la finalidad de obtener mejores resultados.

En este artículo se presenta una alternativa a uno de los operadores utilizados en el proceso de un AG, el operador de mutación. Está demostrado que este operador es necesario para garantizar la convergencia del algoritmo evolutivo. Su objetivo es la modificación del valor de un gen determinado, previamente seleccionado al azar, perteneciente un individuo. Este operador está contralado por el parámetro conocido como probabilidad de mutación, que determina cuándo se aplica dicha mutación.

La importancia del operador mutación ha sido estudiada ampliamente en la literatura, y ha quedado

demostrado que sin él no se garantiza la convergencia del algoritmo [9]. A pesar de todo, el proceso de convergencia puede ser lento, dependiendo de la dificultad del problema. Frecuentemente, esta dificultad implica el uso de cromosomas de tamaño grande, y el tamaño del cromosoma afecta a la eficacia del operador de mutación clásico, que selecciona al azar una posición a mutar: cuanto más largo es el cromosoma, más tiempo será necesario para poder mutar alguna posición concreta del mismo.

En este artículo se presenta una nueva implementación para el operador genético mutación, para problemas en los cuales no existe epistasis entre los genes que conforman el cromosoma de un individuo. Existe epistasis entre los genes de un individuo cuando la expresión de uno o más genes depende de la expresión de otro gen del mismo individuo. Esta interacción sucede cuando la modificación de un gen influye de manera directa sobre el valor en uno o varios genes diferentes del mismo individuo.

La nueva propuesta que se introduce en este trabajo, que como veremos más adelante es una variación de algunos modelos de mutación existente, pretende asociar una probabilidad de mutación a cada gen dentro del cromosoma del individuo. Esta probabilidad se debe ir actualizado con cada mutación sufrida y va a depender de la influencia que dicha mutación pueda tener sobre el valor del fitness global del individuo.

Los resultados presentados en este trabajo demuestran que es posible alcanzar mejores soluciones en un tiempo menor de cómputo, así como utilizando menores generaciones en el proceso evolutivo.

El resto del artículo se divide de la siguiente forma: en la Sección II se detalla un estado del arte de trabajos anteriores donde se aborda el problema de mutación dirigida, la Sección III se describe la metodología utilizada, los resultados de este trabajo se presentan en la Sección IV. Finalmente las conclusiones del mismo se abordan en la Sección V.

II. ESTADO DEL ARTE

La modificación de los diferentes operadores de un AG es un campo ampliamente estudiado por la comunidad científica. Desde el inicio, los investigadores no han dejado de proponer mejoras para los operadores genéticos con la finalidad de obtener AGs que converjan a una solución de forma más rápida. En esta sección nos centraremos en el estudio de posibles mejoras del operador de mutación, fijándonos en particular en versiones del operador que permitan dirigir

su acción hacia zonas específicas del cromosoma que puedan ser de mayor interés.

Una de las primeras mejoras para el operador mutación que se propusieron fue dotar al AG de la capacidad de adaptar la probabilidad de mutación establecida [1], esta técnica es conocida como mutación adaptativa. La idea es simple pero efectiva. Dependiendo del valor de fitness obtenido por los individuos de la población en las diferentes generaciones, la probabilidad se adapta para permitir más o menos mutaciones, de esta manera cuando el fitness global de la población es malo, cuando el proceso de convergencia se estanca, el AG aumentará la probabilidad de mutación, con la esperanza de que estas mutaciones mejoren a los individuos, y viceversa, disminuirá la probabilidad de mutación cuando el fitness de los individuos sea buena para no degradar la calidad de la solución. Esta idea también ha sido aplicada con la probabilidad de cruce. Existen propuestas más avanzadas dentro de la mutación adaptativa, por ejemplo en [2] se propone un operador que modifica la probabilidad de mutación en función del fitness de los individuos de la población, utilizando para ello el algoritmo k-means. Otro enfoque muy utilizado dentro de la mutación adaptativa ha sido la modificación del AG para que en lugar de disponer de una probabilidad de mutación global, cada individuo cuente con una propia. Esta idea se ha codificado tradicionalmente con un gen más al principio del cromosoma y su cálculo se obtiene al generar un número aleatorio dentro de una distribución normal [3]. A lo largo de los años se han utilizado otro tipo de distribuciones de probabilidad, por ejemplo una distribución Levy [4], pero manteniendo la idea anteriormente descrita.

Entre las técnicas utilizadas para mejorar el operador genético de mutación se encuentra la que se conoce como mutación dirigida. Ésta, a diferencia de la mutación adaptativa, basa su funcionamiento en guiar al AG hacia mutaciones más beneficiosas, entendiendo como mutaciones más beneficiosas aquellas que hacen que el valor del fitness de un individuo mejore. En el artículo presentado por Dinabandhu Bhandari et al [7] se propone un operador de mutación que, dependiendo del valor de fitness obtenido en las generaciones anteriores, calcula un punto de mutación para los individuos que integran la población actual.

En el trabajo presentado por Silja Meyer-Nieberg et al.[8] propone una idea novedosa dentro de la mutación adaptativa. Se asocia una probabilidad de

mutación a cada uno de los genes del cromosoma del individuo. Esta probabilidad es un valor aleatorio dentro de un rango. Utilizando esta idea como punto de partida, en este artículo exploramos otras posibilidades, haciendo una primera implementación de una idea similar.

La mutación dirigida que presentamos más adelante, asocia una probabilidad de mutación a cada uno de los genes que componen el cromosoma, en función a la influencia histórica que cada gen ha tenido sobre la mejora de la calidad del individuo, a lo largo del proceso evolutivo. A diferencia del trabajo propuesto en [8], esta propuesta tiene en cuenta cómo ha evolucionado cada individuo de la población, para tratar de guiarlo hacia mutaciones más beneficiosas. El operador que se propone dirige las mutaciones a genes concretos del cromosoma, que mejoren la calidad del individuo, pero en ningún momento modifica el valor de probabilidad de mutación del AG.

III. PROPUESTA DEL NUEVO OPERADOR MUTACIÓN

En esta sección se describe en detalle el nuevo operador genético de mutación propuesto, al cual denominamos *mutación dirigida*, y que se considera en el contexto de problemas de codificación binaria sin epistasias. El objetivo es asociar un valor de probabilidad de mutación a cada gen del cromosoma, y utilizar estos valores de probabilidad en el proceso de mutación. Este nuevo modelo conlleva la utilización de un nuevo vector de probabilidades asociado a cada cromosoma, lo que implicará cambios en el operador de cruce. Estos cambios serán mínimos y solo se utilizan para mantener esta información actualizada en el proceso de cruce.

El nuevo operador de mutación dirigida asigna diferentes probabilidades de mutación a cada uno de los genes de cada individuo de la población, valores que dependerán de la influencia histórica que cada gen ha tenido en la mejora de ese individuo a lo largo del proceso evolutivo.

Para conocer en cada momento la influencia que cierto gen ha tenido sobre la calidad del individuo que lo contiene, es necesario disponer en la codificación del individuo de una nueva estructura capaz de almacenar dicha información. En nuestro caso se ha optado por utilizar un vector, denominado *vector de probabilidad de mutación* (VPM). La Figura 1 muestra un ejemplo gráfico del nuevo cromosoma.

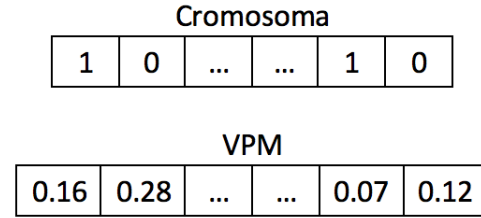


Figura 1. Ejemplo de codificación del VPM

El vector de probabilidades se encarga de almacenar la probabilidad de mutación asociada a cada gen del cromosoma (PMG). La suma de todas las probabilidades contenidas en el VPM debe ser siempre 1, ver figura 1, no pudiendo superar ni disminuir este valor en ningún momento del ciclo evolutivo del AG.

$$\sum_{i=1}^T PMG_i = 1 \quad (1)$$

donde T es el número de genes del cromosoma (Tamaño) y PMG_i es la probabilidad de mutación del gen i -ésimo.

La actualización del VPM es realizada cada vez que un individuo de la población sufre una mutación. Antes de comenzar a explicar en detalle cómo funciona el nuevo operador es importante indicar como se inicializan las PMG del VPM para cada individuo en el proceso de generación de la población inicial. Al generar la población inicial, aún no se tienen datos del histórico de evolución de la fitness, de ahí que todas las posiciones del vector VPM de cada individuo comienzan con la misma probabilidad, esta es calculada según la expresión 2

$$PMG = \frac{1}{T} \quad (2)$$

donde T es el tamaño del cromosoma

III-A. Mutación

Con el vector de probabilidad de mutación inicializado, los pasos a realizar por el nuevo operador de mutación, cada vez que queremos aplicar una mutación a un individuo, son los siguientes:

1. Comprobar si el individuo debe mutar, esta acción viene determinada por el parámetro probabilidad de mutación (P_m), presente en la implementación clásica de cualquier AG.
2. Haciendo uso del método de selección conocido como *ruleta* sobre el VPM del individuo,

se selecciona el gen que sufrirá la mutación. Esta técnica posibilita que cualquier gen pueda ser seleccionado para sufrir una mutación, favoreciendo la selección de aquellos genes con un mayor índice de probabilidad -PMG- en el VPM.

3. Cálculo de la fitness del individuo antes de sufrir la mutación. Hay que recordar, que el proceso de mutación se aplica después del cruce, por lo que los individuos recién cruzados no disponen aún de valor de fitness.
4. Realizar la mutación del gen seleccionado en el paso anterior y cálculo del nuevo fitness del individuo.
5. Actualizar el VPM según el valor resultante, al comparar el fitness del individuo antes de sufrir la mutación y después. Para realizar esta operación se aumentará o disminuirá, una cantidad *alpha* (dependiendo de si la mutación mejoró o no al individuo) la probabilidad asociada a la posición del gen mutado, y proporcionalmente, se disminuirán o aumentará el resto de valores correspondientes al resto de los genes. La suma total de los valores del vector debe seguir siendo uno. Este valor *alpha* será a partir de ahora un nuevo parámetro del algoritmo.

III-B. Cruce

El operador genético de cruce también sufre una modificación con respecto a su implementación clásica, debido a la necesidad de manejar correctamente los vectores de probabilidad de los individuos.

En el operador cruce clásico *one-point crossover* se genera una posición aleatoria del cromosoma que se utiliza para cruzar los dos padres intercambiando material genético en los hijos.

El nuevo operador cruce realiza la misma operación sobre los cromosomas, y además, repite ese mismo proceso con los PMGs de cada padre; pero dado que la suma de los PMGs resultantes de los nuevos individuos no tiene porqué ser 1, al finalizar el cruce es necesario realizar una normalización de estos valores -escalado para que la suma sea 1- para que la suma del nuevo VPM resultante de cada individuo hijo sea 1.

IV. EXPERIMENTOS Y RESULTADOS

En esta sección se describen los experimentos que se han realizado para comparar el nuevo operador genético de mutación que se propone en este trabajo,

con el operador clásico de mutación en problemas sin epistasis entre los genes de los individuos.

Para poder comparar ambos operadores de mutación se ha utilizado un problema bien conocido dentro de la literatura. El problema escogido ha sido el clásico *max-one*. Se trata de un problema de codificación binaria el cual ha sido ejecutado utilizando el operador clásico de mutación y el nuevo enfoque propuesto en este trabajo. La Tabla IV muestra los resultados de los diferentes experimentos que se han realizado. Como puede observarse, estamos particularmente interesados en conocer la utilidad del operador en problemas de tamaño grande: cuando el cromosoma del individuo tiene una longitud tal que son necesarias muchas generaciones para encontrar la solución. Así, hemos probado el operador en tamaños que van de 20 hasta 80 bits.

Para seleccionar un valor *alpha* adecuado para los experimentos, se ha realizado un estudio previo. Con un tamaño de cromosoma igual a 20 bits se han realizado diferentes ejecuciones con diferentes valores para el parámetro *alpha*. De esta manera hemos podido determinar el valor *alpha* que menos generaciones necesitaba para converger a la solución. Como se aprecia en Figura 2 el valor de *alpha* que obtiene mejores resultados es 0.1.

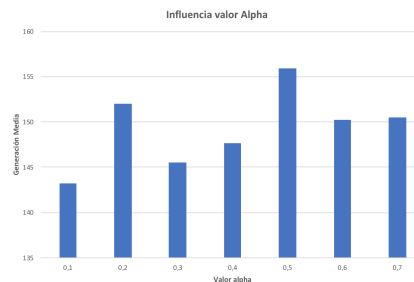


Figura 2. Resultado estudio valor *Alpha*, población 20 bits

Para cada uno de los experimentos que se muestran en Tabla IV se han realizado 150 experimentos. La Figura 3 muestra la generación media que ha tardado el AG en converger a la solución para cada una de las combinaciones mostradas. Como se puede observar el operador de mutación dirigida toma menos generaciones en alcanzar la solución, son necesarias menos generaciones para obtener el mismo resultado que con el operador clásico. Como se aprecia en la gráfica la ventaja obtenida por el nuevo operador respecto al clásico en cuanto al número de generaciones necesarias para obtener una solución aumenta con el tamaño del cromosoma de los individuos de la población.

Experimento	AG	Tam. Población.	Tam. Cromosoma	Num. Generaciones	Prob. Mutación	Alpha
Ex1	C	10	20	20.000	0.01	-
Ex2	I	10	20	20.000	0.01	0.1
Ex3	C	10	40	20.000	0.01	-
Ex4	I	10	40	20.000	0.01	0.1
Ex5	C	10	60	20.000	0.01	-
Ex6	I	10	60	20.000	0.01	0.1
Ex7	C	10	80	20.000	0.01	-
Ex7	I	10	80	20.000	0.01	0.1

Cuadro I
EXPERIMENTOS REALIZADOS

Cuanto el tamaño del individuo es mayor -la dificultad del problema es mayor- la diferencia entre la generación media necesaria para alcanzar solución es mayor entre ambas implementaciones, aportando un mejor rendimiento el operador de mutación dirigida.

La Figura 4 muestra el tiempo medio para alcanzar una solución por ambas implementaciones de los algoritmos. Al igual que sucede con las generaciones, el tiempo medio necesario para obtener la solución por el operador mutación dirigida es menor que la clásica, siendo la diferencia mayor cuanto mayor es el cromosoma del individuo.

Para tamaño de cromosomas superiores a 80 bits ambas implementaciones no siempre convergen a la solución. Aun así, el operador de mutación dirigida converge a la solución un número mayor de veces que el operador clásico (en las 150 ejecuciones realizadas). Para un tamaño de cromosoma igual a 120 bits el operador clásico ha convergido a la solución en 51 ocasiones mientras que el nuevo operador propuesto lo ha hecho 134.

Como decíamos anteriormente, lo aquí mostrado es un análisis preliminar de una variación del operador de *mutación dirigida*. En próximos trabajos pretendemos hacer una comparativa con alguno de los métodos de mutación más cercanos al aquí propuesto, y realizar una selección más adecuada de problemas de test, para estudiar con mayor detalle la potencialidad del nuevo operador. No obstante, estos resultados preliminares muestran el camino a seguir.

V. CONCLUSIONES

Este artículo presenta un análisis preliminar de una nueva versión del operador de mutación adaptado a problemas en los que no existe epistasis. El objetivo de la propuesta es ayudar a determinar en cada momento las posiciones del cromosoma que más pueden ayudar en la mejora de la calidad del individuo tras una mutación.

El nuevo operador asocia una probabilidad de mutación a cada gen del individuo, posición dentro

del cromosoma, que es proporcional a la influencia histórica que ha tenido esa posición sobre la mejora de la calidad del individuo a lo largo del proceso evolutivo. Así, se utiliza un vector de probabilidades VPM, con una posición por cada gen que existe en el cromosoma PMG, para realizar mutaciones dirigidas sobre genes concretos que probablemente beneficiarán la calidad del individuo. Este vector de probabilidades es actualizado en cada proceso de mutación, con un valor que depende de la mejora (o empeoramiento) que la mutación produce en el nuevo individuo cuando se compara con el individuo sin mutación.

Utilizando el clásico problema max-one, de codificación binaria y sin epistasis entre los genes del individuo, hemos comparado el AG clásico con la nueva propuesta que utiliza mutación dirigida, con diferentes configuraciones del problema, variando el tamaño de las poblaciones, el tamaño de los individuos y la probabilidad de mutación del AG. En todos los casos la mutación dirigida ha permitido encontrar soluciones en menor número de generaciones, con una influencia que crece cuando lo hace el tamaño del cromosoma.

Esperamos poder extender este trabajo, con una comparativa más completa con otros operadores de mutación similares, y con un mayor número de problemas, tanto de codificación binaria como real, para entender mejor el impacto del nuevo operador de mutación propuesto.

AGRADECIMIENTOS

Agradecemos el apoyo del Ministerio de Economía y Competitividad proyecto TIN2017-85727-C4-2,4-P, Junta de Extremadura, Consejería de Comercio y Economía, proyecto IB16035 a través del Fondo Europeo de Desarrollo Regional, “Una manera de hacer Europa”.

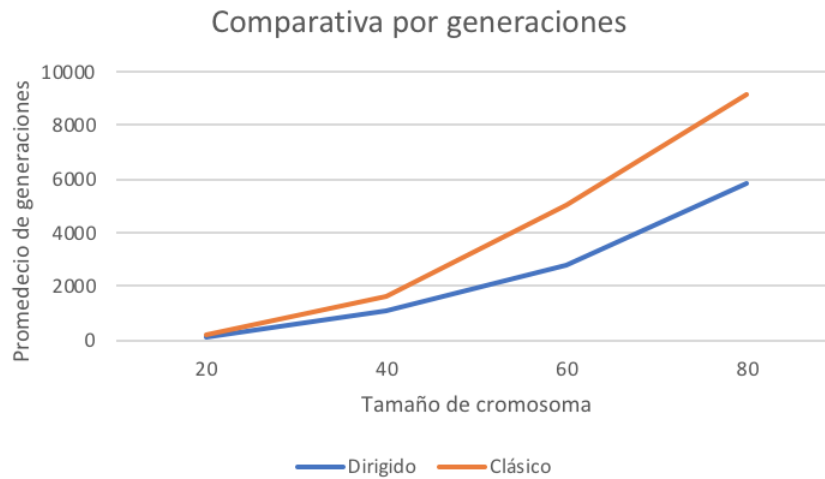


Figura 3. Resultados al comparar las dos implementaciones por número medio de generaciones necesarias para obtener una solución

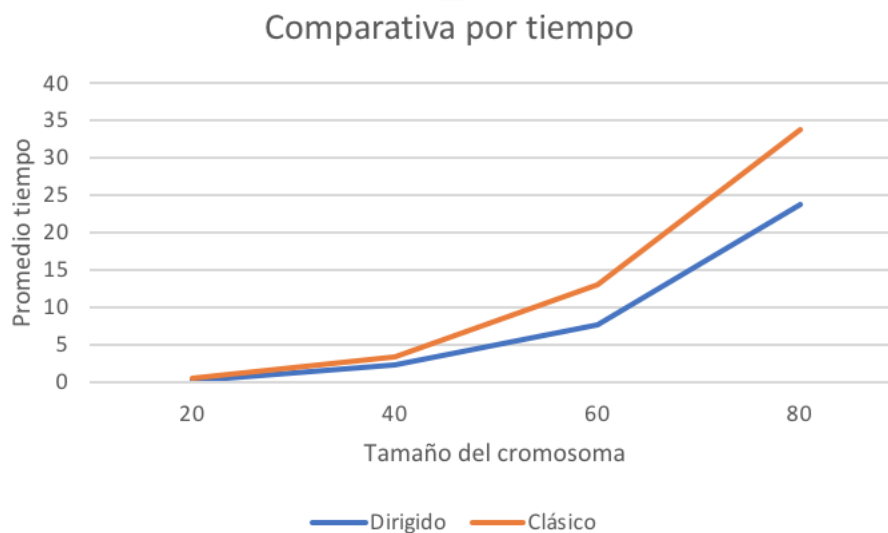


Figura 4. Resultados obtenidos al comparar las dos implementaciones por el tiempo medio necesario para obtener una solución.

REFERENCIAS

- [1] Srinivas, M., Patnaik, L. M. "Adaptive probabilities of crossover and mutation in genetic algorithms," IEEE Transactions on Systems, Man, and Cybernetics, 24(4), 656-667, 1994.
- [2] Zhang, J., Chung, H. S. H., Lo, W. L. "Clustering-based adaptive crossover and mutation probabilities for genetic algorithms," IEEE Transactions on Evolutionary Computation, 11(3), 326-335, 2007
- [3] Fogel, D. B., Atmar, J. W. "Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems," Biological Cybernetics, 63(2), 111-114, 1990.
- [4] Lee, C. Y., Yao, X. "Evolutionary programming using mutations based on the Lévy probability distribution," IEEE Transactions on Evolutionary Computation, 8(1), 1-13, 2004
- [5] Back, T. "Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms," Oxford university press. 1996
- [6] Goldberg, D. E., Holland, J. H. "Genetic algorithms and machine learning," Machine learning, 3(2), 95-99, 1988.
- [7] Bhandari, D., Pal, N. R., Pal, S. K. "Directed mutation in genetic algorithms," Information Sciences, 79(3-4), 251-270. 1994
- [8] Bäck, T. "Self-adaptation in genetic algorithms," In Proceedings of the first european conference on artificial life, 263-271, MIT Press. 1992
- [9] Michalewicz, Z. "Evolution strategies and other methods.," In Genetic algorithms+ data structures= evolution programs (pp. 159-177). Springer, Berlin, Heidelberg. 2016