

Automatización del modelo de simulación de cultivos DSSAT para evaluar el desempeño productivo bajo distintas estrategias de manejo y escenarios ambientales

Figarola, L.A.¹, Ghersa, F.², Castro, R.¹ and Ferraro, D.O.²

¹ Universidad de Buenos Aires (UBA), Facultad de Ciencias Exactas y Naturales, Departamento de Computación. CONICET, Instituto de Ciencias de la Computación (ICC)

lfigarola@dc.uba.ar

rcastro@dc.uba.ar

² Universidad de Buenos Aires (UBA), Facultad de Agronomía, Cátedra de Cerealicultura. UBA-CONICET, Instituto de Investigaciones Fisiológicas y Ecológicas Vinculadas a la Agricultura (IFEVA)

felipeghersa@agro.uba.ar

ferraro@agro.uba.ar

Resumen. La adaptación de sistemas agrícolas a los crecientes cambios ambientales requiere evaluar distintos manejos cuyos resultados brinden el máximo beneficio productivo o ambiental. Los modelos de simulación de cultivos han demostrado ser una herramienta efectiva para llevar a cabo esta tarea. Decision Support Systems for Agrotechnology Transfer (DSSAT) es un software que permite simular el crecimiento, desarrollo y rendimiento de cultivos en base a dinámicas suelo-planta-atmósfera. El objetivo de este trabajo es generar una herramienta de software para explorar rangos de parametrizaciones respecto de las decisiones de manejo (i.e. cantidad y tipo de fertilizante aplicado y secuencias de cultivos), y automatizar la generación de información requerida por DSSAT para simular los modelos. A partir de una lista de cultivos y sus fechas de siembra, se genera una matriz de restricciones para determinar secuencias agronómicamente factibles. Luego, se generan especificaciones estructuradas detallando el clima, suelo, condiciones iniciales, estructura del cultivo y fertilización para cada cultivo. El resultado principal es la exploración de todo el conjunto de decisiones de manejo posibles bajo las mismas condiciones ambientales, permitiendo el análisis y búsqueda de aquellas decisiones que optimicen los beneficios buscados (i.e. rendimiento, rentabilidad económica o conservación de suelo, entre otros).

Keywords: Python; Rotación de cultivos.

1 Introducción

Los cambios en las bases biofísicas de los sistemas de producción (i.e. estabilidad climática, fertilidad de suelos, biodiversidad y disponibilidad energética) ha puesto de manifiesto los desafíos crecientes a los que se enfrentan el sector agrícola [1, 2]. Los

efectos más acuciantes que traen aparejados estos cambios, en las principales regiones productivas, son la caída [3] y creciente inestabilidad [4, 5] en los rendimientos promedio. Asimismo, si se considera que para 2050 será necesario aumentar la producción de fibras alimentos un 60% para abastecer una población de aproximadamente 10 mil millones de personas [6], no es difícil pronosticar un escenario que va a requerir el diseño de sistemas productivos eficientes (i.e. sustentables).

Como consecuencia, será necesario adaptar las estrategias productivas a nuevas condiciones ambientales. No obstante, esto requiere explorar el desempeño de sistemas de producción bajo diversas estrategias de manejo y escenarios ambientales. Los modelos de simulación de cultivos (MSC) utilizan conocimiento de diversas disciplinas como la edafología, climatología, fisiología vegetal y la economía para simular el desempeño ambiental, económico y productivo de distintas combinaciones de suelo, clima y manejo [7]. De este modo, un modelo de simulación adecuadamente validado, calibrado y aplicado puede sobrevenir las limitaciones que imponen los experimentos a campo para diseñar sistemas nuevos a partir de la exploración del espacio total de combinaciones de manejo (i.e. la elección de cultivares, la secuencia de cultivos, y la cantidad y tipo de fertilizantes y pesticidas) [8].

En general, los MSC requieren un trabajo manual intensivo de parametrización y no permiten explorar distintas parametrizaciones para las simulaciones de forma automática. El objetivo de este trabajo es generar una herramienta en para explorar un rango de parametrizaciones respecto de las decisiones de manejo (i.e. cantidad de fertilizante aplicado y secuencias de cultivos), y automatizar la generación archivos requeridos por DSSAT (Decision Support System for Agrotechnology Transfer) para correr sus modelos [9].

2 Materiales y Métodos

Para la simulación de cultivos se utilizó el simulador DSSAT versión 4.7.5 [9], el cual permite simular el crecimiento, desarrollo y rendimiento de cultivos en base a dinámicas suelo-planta-atmósfera. En particular, DSSAT utiliza información acerca del sitio, clima diario durante el período de crecimiento, características edáficas al principio del ciclo de cultivo o rotación y decisiones de manejo. Estas decisiones incluyen a la estructura del cultivo, labranza, riego y fertilización. El programa es de acceso libre para uso académico a través del sitio <https://DSSAT.net>.

Los scripts para la automatización fueron realizados en Python 3 [10]. Para ejecutar DSSAT en línea de comando se utilizaron los módulos *so* y *subprocess*. Para leer la matriz de restricciones se utilizó el módulo *csv*. Se utilizó el módulo *os.path* para trabajar con rutas de forma general.

3 Resultados: Estructura de Automatización

A partir de una lista de cultivos $\mathcal{C} = \{C_1, \dots, C_c\}$ se generan todas las combinaciones o secuencias $S = \{S_1, \dots, S_s\}$ de 1 a 4 cultivos sucesivos. Por razones agronómicas y de uso del suelo, es necesario generar reglas que restrinjan el conjunto de secuencias

que serán simuladas. De este modo, un conjunto de restricciones sobre la generación de secuencias es la imposibilidad de sembrar ciertos cultivos mientras que haya otro cultivo sin cosechar en el lote simulado. Por ejemplo, en la localidad de Pergamino, el maíz se siembra a mediados de septiembre, mientras que el trigo se cosecha hacia fines de noviembre. Como consecuencia, surge una imposibilidad de sembrar el maíz como cultivo sucesivo a trigo. La Figura 1 muestra un grafo que representa las transiciones factibles en función de la fecha de siembra y fecha de cosecha para un conjunto de cultivos teóricos C_c .

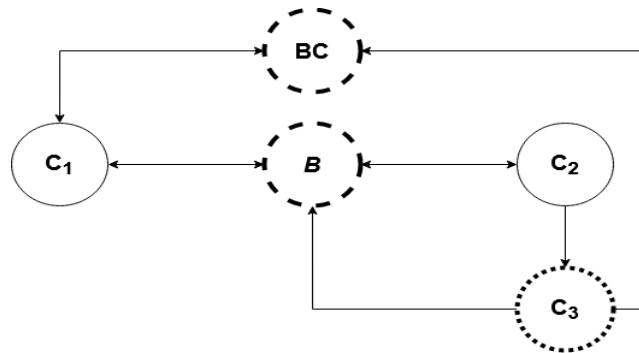


Fig. 1. Grafo de restricción para generar secuencias agrónomicamente factibles para cultivos estivales e invernales. La dirección de las flechas indica secuencias realizables entre dos estados de ocupación de la tierra. Los círculos de línea llena (C_1 de verano y C_2 de invierno) indican cultivos de primera. El círculo de línea punteada (C_3) indica un cultivo de segunda. Los círculos de línea de trazos gruesa (BC y B) indican barbechos con y sin cobertura vegetal, respectivamente.

En función de las secuencias seleccionadas, utilizando los criterios de factibilidad, se procede a construir estructuras de datos necesarias para ejecutar DSSAT. En la Figura 2, se presenta el flujo de información para la automatización de DSSAT que comienza con la generación de secuencias agrónomicamente factibles. Luego, la aplicación requiere que se defina el sitio (i.e. clima y suelo) así como los cultivares y las variables de manejo mediante un conjunto de archivos de texto. Dichos archivos alimentan el módulo principal de simulación, que luego predice un conjunto de variables de desempeño productivo y ambiental como el rendimiento, el uso del agua o el uso y lixiviación de nutrientes. De este modo, es necesario generar los coeficientes genéticos de cultivares para las especies seleccionadas en las secuencias, las variables diarias de las series climáticas y las características de los perfiles del suelo. Estos archivos no serán modificados en la automatización de modo que pueden ser generados de forma manual o utilizando a la interfaz gráfica que provee DSSAT (i.e. *GenCalc* para coeficientes genéticos de cultivares, *Weatherman* para series climáticas y *SBuild* para características del suelo).

A partir de esta información, se procede con la generación de un archivo *.SQX* que DSSAT toma como entrada para simular en modo de secuencia. A diferencia del modo de experimento o simulación de temporada, la secuencia simula las condiciones

iniciales del cada cultivo de manera sucesiva, de modo que utiliza los valores de salida de condiciones del suelo en tiempo $t - 1$ para simular el cultivo en tiempo t . Este archivo suele generarse manualmente o mediante la interfaz gráfica de DSSAT, *XBuild* [9]. En el archivo *.SQX* es necesario especificar las características de los perfiles del suelo, la serie climática y los cultivares de cada especie que serán utilizados en la simulación. Asimismo, se deben indicar detalles de siembra para cada cultivo (i.e. fecha, densidad, profundidad, y distribución, dirección y distancia entre hileras). Por último, es necesario indicar los detalles de la fertilización (i.e. fecha, producto, método, profundidad y cantidad).

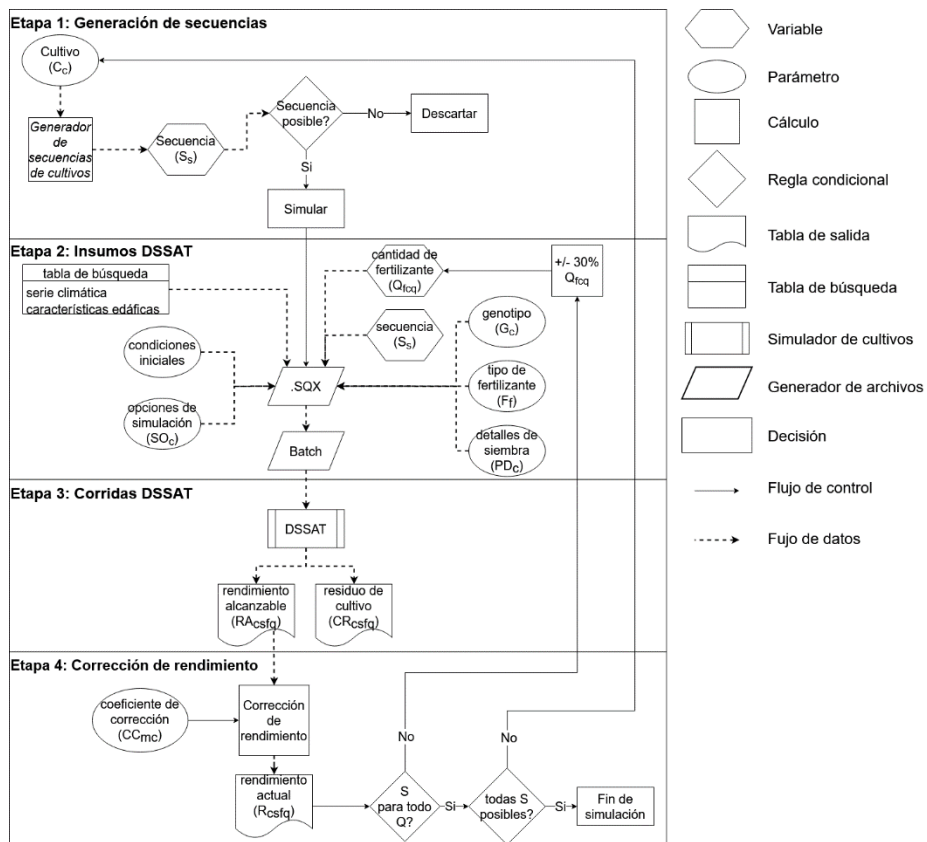


Fig. 2. Diagrama de flujo para la automatización de DSSAT.

Dado que se quiere evaluar el desempeño de distintas combinaciones de variables de manejo (en particular, secuencias de cultivo y niveles tecnológicos definidos por cantidad y tipo de fertilizante y fitosanitarios aplicados) mediante scripts de Python se modificará la secuencia de cultivos S_s y la cantidad de fertilizante aplicado Q_{fc} , donde $q = 1, \dots, Q$ es la cantidad correspondiente a nivel tecnológico bajo, medio o alto, del tipo de fertilizante $f = 1, \dots, F$, para el cultivo $c = 1, \dots, C$. La secuencia

simulada será modificada a partir de la matriz de tratamientos en el archivo *.SQX*. La cantidad de fertilizante aplicado será modificada utilizando un criterio de $\pm 30\%$ a partir del nivel medio definido en el archivo *.SQX* inicial para cada cultivo simulado [11]. Este valor se puede modificar en función de los requerimientos de simulación. Luego, para ejecutar DSSAT desde línea de comando, se generó un script para crear un archivo *Batch* correspondiente a la secuencia simulada (ver código en la Sección Acceso).

Por último, dado que DSSAT simula el rendimiento alcanzable (i.e. limitado por agua y nutrientes), se definieron coeficientes de corrección del rendimiento para obtener un valor estimado del rendimiento actual (i.e. limitado por recursos y efectos de adversidades biológicas). Estos coeficientes se definieron en función del nivel tecnológico aplicado, simulando el efecto de la aplicación de fitosanitarios sobre el rendimiento.

4 Conclusión

Se presentó un esquema de automatización del modelo de simulación de cultivos DSSAT mediante el uso de scripts de Python. Dicha automatización constituye un avance clave para aumentar la capacidad de evaluación del desempeño de sistemas de producción mediante este tipo de modelos de simulación a partir de la automatización de distintas secuencias de cultivos y paquetes tecnológicos.

El esquema presentado sirve para explorar el desempeño de distintas estrategias de manejo agrícola. En particular, está configurado para evaluar el rendimiento del cultivo, aunque DSSAT permite simular otras variables como la materia orgánica del suelo o la salida de nitrógeno del sistema. De este modo, se puede extender la utilidad de la automatización a otras variables de desempeño como el análisis económico (i.e. margen bruto o rentabilidad) o esquemas de desempeño ambiental como la síntesis emergética (con M) [12].

Asimismo, mediante el acople del nuevo esquema de automatización con algoritmos de optimización es posible utilizar encontrar compromisos entre las variables de desempeño ambiental y económico [13, 14]. El aporte fundamental de este trabajo es una herramienta que permite a los usuarios de DSSAT potenciar su capacidad de explorar escenarios productivos, reduciendo los costos de tiempo y errores asociados a la generación manual de los escenarios. Este tipo de ejercicio exploratorio resulta fundamental para adaptar los sistemas de producción a los cambios en las condiciones ambientales para asegurar un uso eficiente y sustentable de los recursos y demuestra la potencialidad de la interacción entre el manejo agronómico y las ciencias de datos.

5 Acceso

Código completo y prueba de concepto:

<https://github.com/lucasfigarola/automatizacion-de-DSSAT>

Referencias

1. Andrade, F.H., et al., *Los desafíos de la agricultura argentina: satisfacer las futuras demandas y reducir el impacto ambiental*. 2017: Ediciones INTA.
2. López, S.L., et al., *Pesticides used in South American GMO-based agriculture: A review of their effects on humans and animal models*, in *Advances in Molecular Toxicology*. 2012, Elsevier. p. 41-75.
3. Challinor, A.J., et al., *A meta-analysis of crop yield under climate change and adaptation*. *Nature Climate Change*, 2014. **4**(4): p. 287-291.
4. Powell, N., et al., *Yield stability for cereals in a changing climate*. *Functional Plant Biology*, 2012. **39**(7): p. 539-552.
5. Garibaldi, L.A., et al., *Global growth and stability of agricultural yield decrease with pollinator dependence*. *Proceedings of the National Academy of Sciences*, 2011. **108**(14): p. 5909-5914.
6. Alexandratos, N. and J. Bruinsma, *World agriculture towards 2030/2050: the 2012 revision*. 2012, FAO: Rome.
7. Jones, J.W., et al., *Brief history of agricultural systems modeling*. *Agricultural systems*, 2017. **155**: p. 240-254.
8. Royce, F., J. Jones, and J. Hansen, *Model-based optimization of crop management for climate forecast applications*. *Transactions of the ASAE*, 2001. **44**(5): p. 1319.
9. Hoogenboom, G., et al., *Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.7.5 (<https://DSSAT.net>)*. 2019, DSSAT Foundation: Gainesville, Florida, USA.
10. Van Rossum, G. and F.L. Drake, *The python language reference manual*. 2011: Network Theory Ltd.
11. Ferraro, D.O. and M. Gagliostro, *Trade-off assessments between environmental and economic indicators in cropping systems of Pampa region (Argentina)*. *Ecological indicators*, 2017. **83**: p. 328-337.
12. Odum, H.T., *Environmental accounting: emergy and environmental decision making*. 1996: Wiley New York.
13. Groot, J.C., G.J. Oomen, and W.A. Rossing, *Multi-objective optimization and design of farming systems*. *Agricultural Systems*, 2012. **110**: p. 63-77.
14. Kropp, I., et al., *A multi-objective approach to water and nutrient efficiency for sustainable agricultural intensification*. *Agricultural Systems*, 2019. **173**: p. 289-302.