Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática
MA55G - Laboratorio de Modelamiento Matemático I

Informe Final

Adaptación de NSGA-II para optimizar el plan de observaciones de un telescopio

Alumno: David Hasson

Investigador: Francisco Förster

Profesores: Alejandro Jofré
Alejandro Maass
Jaime San Martín

2 de enero de 2015

Índice

1. Introducción

Un desafío importante en la determinación del plan de observaciones de un observatorio es la necesidad de satisfacer distintos criterios científicos de forma simultánea. Dependiendo del proyecto y de los distintos grupos científicos que comparten el uso de las instalaciones estos criterios podrían incluso ser conflictivos. Una manera de abordar la elección del plan de observaciones consiste en modelar el problema como uno de optimización multiobjetivo. En este caso, encontrar un plan de observación eficiente consiste en encontrar soluciones no dominadas.

El siguiente reporte da cuenta del trabajo realizado por el autor durante el segundo semestre del año 2016 en el Laboratorio de Astroinformática del Centro de Modelamiento Matemático. El objetivo fue el diseño de un algoritmo para construir planes válidos, modelar el desempeño de planes de observación dados y la adaptación de un algoritmo genético para una versión simplificada del problema considerado. Los métodos usados fueron programados en Python. Se realizaron algunas simulaciones computacionales, identificando las dificultades que se presentaron. Se presentan los resultados de las simulaciones realizadas con el objetivo de analizar el alcance y validez del algoritmo principal. Por último se analizan los resultados obtenidos. Se indican posibles caminos de mejoras para disminuir el tiempo de ejecución del algoritmo que podrían servir al momento de aplicarlo o extenderlo.

2. Descripción del problema

El objetivo principal de este trabajo fue generar una aproximación para la optimización del plan de observaciones del proyecto LSST (*Large Synoptic Survey Telescope*). Éste constará en un telescopio de gran campo angular ubicado en Cerro Pachón, realizando observaciones del cielo durante 10 años. Se quiere definir un plan de observaciones para el telescopio para todas las noches, y se desea cumplir una serie de objetivos científicos de manera simultánea.

Como se empleará un telescopio robótico, funcionará de manera autónoma. Por lo tanto, un plan de observaciones se debe poder codificar e interpretar de manera que el telescopio ejecute el plan deseado. Se espera que los datos recolectados a lo largo de los 10 años ayuden a responder una amplia gama de preguntas en áreas como la astrofísica, la cosmología y la física.

En la práctica, durante el proceso de observación astronómica deben cumplirse ciertas condiciones:

- La región del cielo donde es posible apuntar el dispositivo está restringida.
- No es posible observar a un ángulo menor que 45° del horizonte del observador.
- Las observaciones deber estar a más de 40° de la Luna.
- Las observaciones se realizan en pares de 15 segundos de exposición.
- El telescopio puede observar en distintas frecuencias usando filtros diferentes, pero cada cambio de filtro toma 2 minutos.
- El Sol debe estar a más de 18° del horizonte.

En este trabajo, por tratarse de una primera aproximación al problema, no se consideraron los últimos 2 requisitos. Se considerará en lo que sigue que el cielo está dividido en distintas regiones disjuntas tal que cada una puede ser cubiertas por el lente del telescopio en una observación. Adicionalmente, en la planificación de 10 años se desea realizar al menos 1000 observaciones por cada una de las regiones del cielo observable (notar que la porción observable va variando en el tiempo), lo cual se puede aproximar dándole alta prioridad al tiempo transcurrido desde la última vez que fue visitado. Una región se dice factible si cumple los requisitos mencionados anteriormente. Siguiendo la línea, un plan de observaciones se dice factible si cada una de las observaciones que realiza es factible en el momento que es realizada. En particular, en un plan factible el tiempo de cada observación será de 30 segundos.

Una dificultad que se presenta al modelar el problema es que tanto las restricciones como las funciones objetivo dependen del momento de observación. Esto no imposibilita el trabajo, pues dada la ubicación del observatorio en la Tierra existen herramientas computacionales para calcular los puntos factibles del cielo en cada instante. No se entrará en detalles técnicos, sólo para completar la descripción mencionamos que se trabajó en base a 2 sistemas de referencia: el de coordenadas horizontales (azimut-altitud) que depende de la ubicación del observador y del momento, y el de coordenadas ecuatoriales (ascensión recta - declinación), que se pueden considerar fijas durante los 10 años y por tanto sirven para identificar los puntos de interés en el cielo a lo largo del tiempo. Se conocen fórmulas para realizar la conversión de un sistema al otro.

Otra dificultad es que no hay expresiones cerradas que expresen los objetivos científicos del proyecto. Lo que sí se sabe es que los objetivos dependen, respectivamente, de:

- La calidad del plan de observaciones, vista como calidad de las imágenes obtenidas.

- El tiempo entre visitas sucesivas a cada *target*.[1]

La calidad de cada observación depende del ángulo cenital asociado,[2] mientras menor sea este ángulo mejor será la calidad de la imagen. El segundo objetivo tiene que ver con que las observaciones se realicen regularmente y se genere una cantidad similar de datos para cada *target*. La meta es entonces minimizar el primer objetivo y maximizar el segundo. Es importante notar que el LSST puede observar el cielo visible completo en pocas noches, esto se debe a su cámara capaz de cubrir un área equivalente a 40 veces el tamaño de la Luna llena [7].

Un detalle no menor es que el plan de observaciones puede ser afectado por dificultades ambientales no predecibles. Un ejemplo típico son las nubes, llegando a imposibilitar el proceso. Se debe tomar una decisión de qué hacer en estos casos, omitir las observaciones de la noche perdida puede ser mala idea, y por otro lado, si se realiza un nuevo proceso de planificación lo ideal sería que tome poco tiempo de ejecución, debido al corto plazo disponible para tomar la decisión. Se denominará problema estático al caso en que no se considera este factor, y problema dinámico al caso en que sí. Otra opción para enfrentar el caso dinámico puede ser modelar el surgimiento de nubes con un pronóstico en base al historial de años anteriores, incorporando noches sin observación como parte del plan.

3. Resolución

Dados los puntos anteriormente explicados, los pasos a seguir son los siguientes:

1. Construir planes de observación válidos.
2. Modelar el desempeño de planes de observación dados.
3. Utilizar algoritmos heurísticos para optimización (en este caso, algoritmo genético).
4. Estudiar impacto y factibilidad de optimización en distintas escalas de tiempo (minutos, horas, días, etc.) para el caso estático. En particular, analizar la extensión del enfoque para el caso dinámico.

En las siguientes páginas se describe el trabajo realizado respecto a cada una de las etapas. Cabe mencionar que los puntos 1 y 2 se realizaron en conjunto con A. Cristi, pues para ambos trabajos servía esta rutina inicial y además se consideró importante lograr construir planes de buena calidad, tarea a la que se destinó bastante tiempo. En lo que sigue, *buena calidad* se refiere a soluciones que tengan relativamente bajo valor en el objetivo asociado a calidad de imagen y relativamente alto valor en el objetivo asociado a tiempo entre visitas. En dicha fase fue crucial conocer lo que ha funcionado bien en otros proyectos recientes, lo cual fue aportado por el investigador del CMM a cargo de supervisar el trabajo. Ver por ejemplo la sección 2.2 de [4]. En dicho caso se sesgó la elección del nuevo *target* observado dando preferencia a un menor tiempo de movimiento del telescopio desde el punto de observación actual. [3]

[1] Se denomina *target* a una región del cielo en la discretización considerada.

[2] El ángulo cenital de un objeto en el cielo corresponde al ángulo entre la posición sobre el observador y el objeto. Por ejemplo, el ángulo cenital del objeto será 0° si éste está directamente sobre la cabeza del observador, y 90° si el Sol se sitúa sobre el horizonte.

[3] Así, entre 2 objetos de misma calidad se estará priorizando al que el telescopio demora menos en llegar, y entre 2 objetos que fueron visitados por última vez hace la misma cantidad de días se preferirá también el que se demora menos en llegar.

3.1. Creación de planes factibles

Como ya se mencionó, la primera fase consistió en el diseño e implementación de una rutina para generar planes de observación factibles de buena calidad. Adicionalmente fue creado un método de visualización del plan de observaciones en un plano esférico altitud-azimut. A nivel de cálculo de ángulos cenitales, la noche se discretizó en 10 intervalos de igual largo en que se considera que éstos ángulos no cambian. Notar que al transcurrir una hora corresponde a movimiento de 15° en el cielo. En el futuro se pueden usar discretizaciones más finas buscando el equilibrio entre tiempo de cálculo y precisión.

Se debían crear criterios de elección de buenos puntos factibles para ser incorporada a un algoritmo iterativo que va incorporando observaciones a un plan (codificado como arreglo) y avanza en el tiempo de observación (30 segundos) más el de desplazamiento del punto actual al siguiente, simulando el proceso de un observatorio. La primera propuesta fue generar, en cada instante de elección, 2 rankings de las regiones del cielo, uno según orden creciente de ángulo cenital y otro según orden decreciente de tiempo desde la última visita (sin ordenar el arreglo original). Luego, combinarlos según ciertos coeficientes que resultaran en planes razonables en la práctica (para dicho análisis se usó la herramienta que grafica el plan a lo largo de la noche).

Posteriormente, notando que no es necesario realizar los cálculos anteriores sino que basta calcular una combinación (con pesos) de los valores de ambos arreglos, se cambió el criterio de elección por el siguiente:

- Se construye el arreglo
$$\frac{((T_{Fact} \geq 3{,}65) + 0{,}01) \times 1000 T_{Fact}}{5|Z(Fact, Night_{discrete})| + 5Dist + 1}$$
donde
  - $(T_{Fact} \geq 3{,}65)$ denota una arreglo de valores binarios que valen 1 si la desigualdad se cumple y 0 si no.
  - T es el arreglo con el tiempo transcurrido desde la última visita.
  - $Fact$ es el conjunto de índices de regiones factibles en el cielo en el instante de elección.
  - $Z(Fact, Night_{discrete})$ es el ángulo cenital en cada punto de $Fact$, en el instante $Night_{discrete}$ correspondiente a uno de los 10 en que se discretiza el cálculo de ángulos cenitales.
  - $Night_{discrete}$ corresponde al intervalo actual en que se encuentra la observación.
  - $Dist$ es un arreglo con las distancias (en tiempo de movimiento del telescopio) desde la observación actual a $Fact$.

  Notar que la primera expresión va a tomar el valor 1 si han transcurrido más de 3.65 días desde la última observación al *target*, y 0 si no (el 3.65 se fijó para imponer una tasa de visitas, debido a que se quiere visitar cada punto 1000 veces en aproximadamente 3650 noches). El valor 0.01 se agregó para que no haya ceros que podrían causar que se descarte una región que según los otros criterios sí se desearía agregar.

- Encontrar los índices de los 6 mejores valores del arreglo anterior, lo cual se puede hacer en orden lineal en su tamaño.

- Elegir al azar uno de los índices anteriores. Esta será la observación que se añade.

De esta manera, en cada instante se prioriza agregar los puntos factibles asignando pesos a su ángulo cenital actual y tiempo desde última visita, ponderado por una función decreciente en el tiempo que demora el cambio de posición del telescopio

3.2. Funciones Objetivo

En base a la discusión anterior, se propusieron las funciones:

1. $\sum_{i \in \mathcal{P}} z_i(t)$
2. $\sum_{i \in \mathcal{P}} {\Delta t_i}^2$

donde

- $\mathcal{P}$ es un plan de observaciones
- $z_i(t)$ es el ángulo cenital del objetivo i en el tiempo t que es visitado por $\mathcal{P}$
- Δt_i es el tiempo en días desde la observación anterior al objetivo i en el momento que éste es visitado nuevamente.

Al buscarse minimizar la primera función y maximizar la segunda, éstos indicadores permiten modelar los objetivos científicos subyacentes.

Adicionalmente se programó una tercera función a minimizar, asociada al número de visitas por *target*:

$$\sum_{i \in \text{targets}} V_i(\mathcal{P})$$

donde $V_i(\mathcal{P})$ vale 1 si $\mathcal{P}$ visita menos de N_{REF} veces i, y 0 sino. En el caso de planificación a 10 años $N_{REF} = 1000$. Lo deseable sería alcanzar planes de observación óptimos tales que el tercer objetivo valga 0. Como se trabajó a escala de una sola noche, esta función no fue utilizada en las simulaciones.

Posteriormente, la segunda función objetivo fue cambiada por $\sum_{i \in \mathcal{P}} 2^{\Delta t_i}$. Con esta expresión se da aún más prioridad a que distintas visitas a un mismo *target* sean realizadas en días diferentes.

3.3. Algoritmo de optimización

Antes de pasar a la descripción e implementación, se presenta una breve descripción del contexto en que se trabaja. Se sigue la notación de [6]. Sea F el vector de funciones objetivo. Supongamos sin pérdida de generalidad que se desea minimizar cada una de las funciones que componen F. Un problema de optimización multiobjetivo tiene la forma

$$\begin{aligned} \text{mín} \quad & F(x) = (f_1(x), f_2(x), ..., f_n(x)) \\ \text{s.a.} \quad & x \in S \end{aligned}$$

donde $n \geq 2$ es el número de objetivos y S es el conjunto de soluciones factibles.

Dado que los objetivos podrían ser conflictivos (como es el caso en nuestro problema: no necesariamente existirá un plan de observaciones que maximice la calidad y al mismo tiempo minimice la función de los tiempos entre visitas, cada una por sí sola), se trabaja con una nueva noción de optimalidad, o eficiencia:

Definición. *Dados dos vectores $u = (u_1, ..., u_n)$ y $v = (v_1, ..., v_n)$ en el espacio de valores de las funciones objetivo, se dirá que u domina a v si y sólo si ninguna componente de v es menor que la correspondiente en u y por lo menos una componente de u es estrictamente menor que la de v, es decir:*

$$u_i \leq v_i \; \forall i \in \{1, \ldots, n\} \wedge \exists i \in \{1, \ldots, n\} : u_i < v_i.$$

Definición. *Una solución factible x es Pareto-óptima si no existe una solución y tal que $F(y)$ domina a $F(x)$. Es decir, es una solución no dominada. La imagen del conjunto de soluciones no dominadas se denomina Frente de Pareto.*

Las ideas anteriores se extienden de forma directa al problema general con maximización y/o minimización de funciones objetivo. Notar que a priori el frente de Pareto es desconocido y es posible que la única manera de conocerlo para el caso de planificación de observaciones sea una búsqueda exhaustiva. Además, no necesariamente es un conjunto convexo (o cóncavo) o continuo.

Dado lo anterior, la meta es aproximar el frente de Pareto, cubriendo una parte amplia de éste para tener un espectro abundante de decisiones posibles. Se usó NSGA-II (Nondominated Sorting Genetic Algorithm II), un algoritmo genético para optimización multiobjetivo. Tanto en su forma original como levemente modificado, ha sido aplicado de manera exitosa en múltiples áreas. Además, se optó por usar NSGA-II dados los resultados satisfactorios que se han presentado en otros problemas de observación astronómica [4, 5].

A continuación se presenta una breve descripción del algoritmo usado. Los algoritmos genéticos son metaheurísticas para enfrentar problemas de optimización [3, 6]. En este caso, lo que se realiza es operar sobre una **población** de N **individuos** (conjunto de planes de observación factibles).[4] La idea es operar en cada etapa con las soluciones actuales, en cada **generación** (iteración) se realiza lo siguiente:

- Seleccionar según criterios de índice de no-dominación. La potencia de NSGA-II está en que genera capas de frentes según no-dominación (El primero sería el de soluciones no-dominadas, después viene el de las que sólo son dominadas por soluciones del primer frente, y así sucesivamente hasta terminar).
- Luego, se opera sobre la población actual generando **descendientes**: Se usan los operadores de **crossover** y **mutación** que serán descritos posteriormente.
- Se eligen los mejores N individuos entre la población y los descendientes, según el proceso de generación de capas de no-dominación y priorizando la selección de soluciones que están alejadas (relativamente) de las restantes, para mantener diversidad en la población.
- El conjunto de soluciones no dominadas al final de la última iteración será el frente de Pareto que retorna el algoritmo.

A continuación se explican los operadores genéticos basados en una aplicación previa [5].

- **Crossover:** Dados 2 planes de observaciones, con probabilidad p_c se realiza lo siguiente: Elegir un tiempo T_c aleatorio durante la noche y dividir ambos planes en el segmento antes y después de T_c. Luego, se pega la primera parte de uno con la segunda del otro, generando 2 individuos. p_c se denomina *probabilidad de crossover*, y en las aplicaciones típicamente toma valores entre 0.3 y 0.9.
- **Mutación:** Luego de aplicar la fase de *crossover*, para dado un plan de observaciones entre los descendientes se elige una observación al azar entre las que lo componen. Ésta se cambia por otro punto que sea factible en su instante de observación.

[4] Los planes iniciales en este caso son sesgados, dado que en experiencias anteriores si se inicializa con planes arbitrarios el algoritmo puede tardar demasiado tiempo en traer resultados deseables, en cambio, cuando la población inicial se sesga se debe introducir suficiente aleatoriedad durante el proceso posterior para evitar una convergencia temprana fuera del frente de Pareto.

Los operadores anteriores se aplican para diversificar la población de modo de tener una idea más general de cómo se comportan las funciones objetivo a lo largo del conjunto factible.

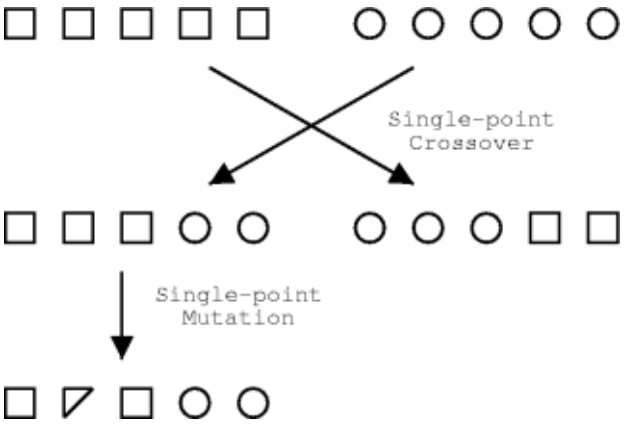


Figura 1: Ilustración del funcionamiento de los operadores genéticos. Imagen tomada de `http://gaul.sourceforge.net`.

Un problema que surge al momento de aplicar mutación y *crossover* es que al cambiar un punto de observación a lo largo de la secuencia cambian los tiempos de observación posteriores, así como los de traslación del telescopio, y eventualmente la factibilidad de las observaciones posteriores. Analogamente, al cruzar 2 planes de observación puede generarse una ventana temporal o un traslape de observaciones (con respecto al tiempo de observación que cada *target* tiene en su *schedule* original. Los operadores deben trabajar con planes factibles, luego, debe tomarse una decisión para repararlos. En este momento se evaluó el uso de varias rutinas de **reparación**, siendo una etapa compleja debido a su efecto sobre el desempeño del algoritmo. Se determinó usar un criterio relativamente optimista:

- Si el tiempo de la observación en el punto de mutación o cruce es cercano al tiempo que tenía la observación siguiente en el plan original (con una tolerancia de 30 segundos), no modificar.
- Si no lo es: recorrer las observaciones desde el punto conflictivo en adelante, incluyendo el *target* respectivo si resulta factible en el nuevo instante que le toca, o eliminándolo del plan si no. Ésto puede dejar tiempo muerto al final del plan, por lo que éste se rellena con observaciones hasta completar la noche. Para ello se ocupa la misma heurística de inicialización.

La idea detrás de la tolerancia de 30 segundos es que, como el tiempo de observación es de 30 segundos, no se estaría perdiendo la opción de incluir otra observación entre medio en caso que quede un *gap* temporal. Si queda un traslape entre observaciones, éste sólo podrá ser pequeño y se intentará arreglar a posteriori.

Para utilizar el algoritmo NSGA-II se trabajó con el paquete DEAP[5] que tiene implementados distintos objetos y módulos que vuelven personalizable el algoritmo genético. Además, la arquitectura de DEAP permite incorporar herramientas de computación distribuida, como el módulo `multiprocessing` de Python o la librería SCOOP [6]. Además, DEAP cuenta con distintas herramientas que sirven de apoyo al momento de analizar el desempeño del algoritmo. Para una exposición consisa pero detallada sobre DEAP y su modo de uso, el lector puede revisar el documento [2].

[5]Del inglés *Distributed Evolutionary Algorithms in Python*. Más información en https://github.com/deap.
[6]Del inglés *Scalable COncurrent Operations in Python*. Más información en http://pyscoop.org.

3.4. Implementación: detalles técnicos

Todo fue programado en el lenguaje Python. Las simulaciones se realizaron en la versión 2.7.10 para 32 bit un computador de escritorio con 8 GB de RAM y procesador de 2.16 GHz de 4 núcleos. A continuación se presenta un resumen de los paquetes utilizados.

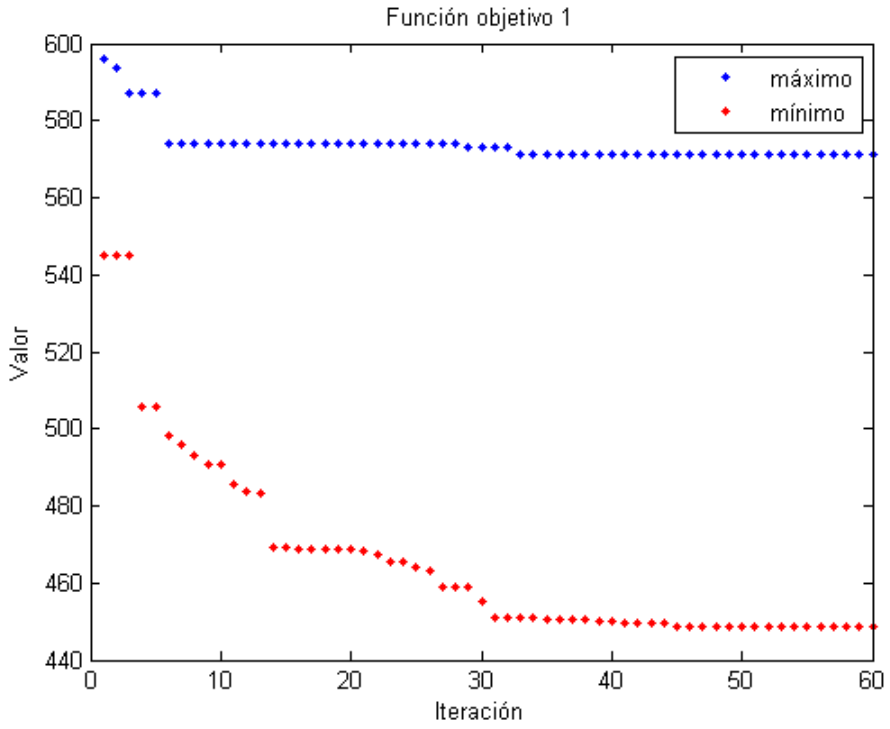
Paquete	**Función**
PyEphem	Incorporar elementos y cálculos astronómicos.
NumPy	Trabajar con arreglos y simular variables aleatorias.
DEAP	Estructuras base para el algoritmo genético.
SCOOP	Distribuir tareas concurrentes.
matplotlib	Paquete estándar para graficar.

Tabla 1: Breve descripción de los paquetes utilizados.

Para la discretización del cielo se usó el resultado del algoritmo HEALPix (*Hierarchical Equal Area Iso Latitude pixelation of the sphere*). El tiempo desde la última visita se inicializa para cada *target* como un entero entre 1 y 4 días. Se simuló un observador ubicado en latitud -33:27:00, longitud -70:40:00 y fecha 03/12/2015 con tiempo de inicio de observaciones a las 23:00:00.

4. Resultados

Se ejecutó el algoritmo genético por 60 generaciones, con población de $N = 100$ individuos y $p_c = 0.5$. El último parámetro fue elegido luego de distintos experimentos, con un valor más bajo o muy alto el algoritmo se estancaba rápidamente, casi sin obtener mejoras en la exploración a partir de un punto (manteniendo la población básicamente fija, lo cual se puede observar al analizar el máximo, mínimo, promedio y desviación estándar de las funciones objetivo en la población al avanzar las generaciones). El tiempo de ejecución fue de entre 2.5 y 3 horas, de las cuales la fase de inicialización tomó alrededor de 1.5 hrs. Los siguientes gráficos resumen la información obtenida.



1.0

Figura 2: Evolución del valor máximo y mínimo de la primera función objetivo a lo largo de las iteraciones.

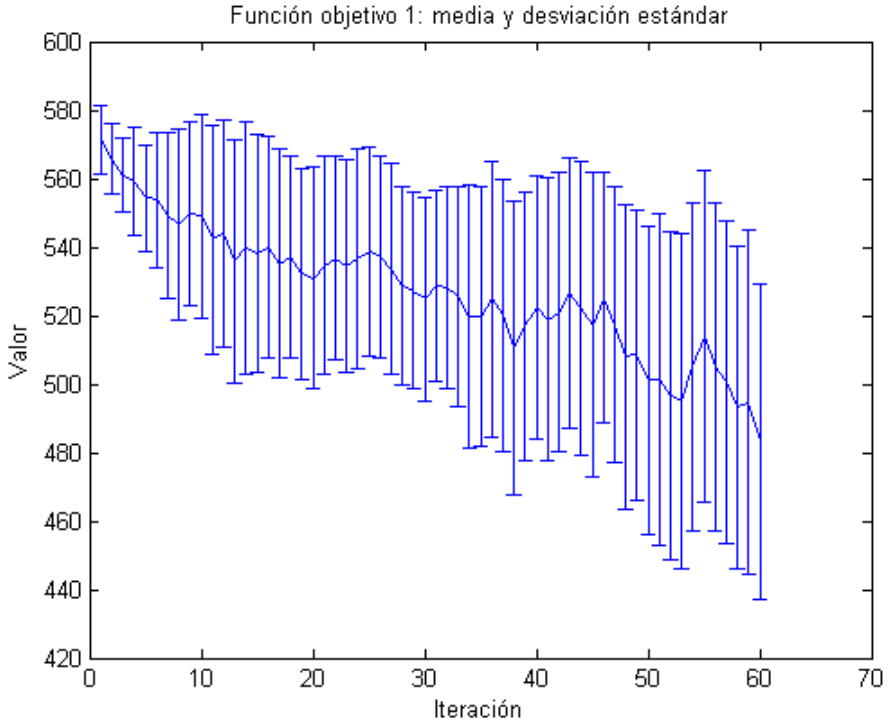


Figura 3: Evolución de la media y error (desviación estándar) del valor de la primera función objetivo en la población.

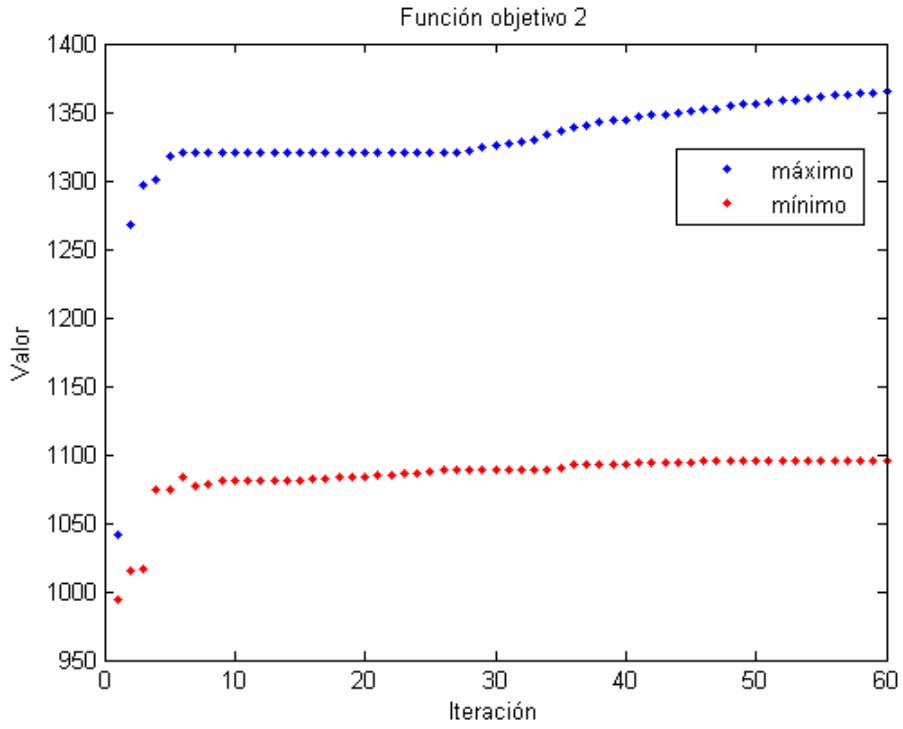


Figura 4: Evolución del valor máximo y mínimo de la segunda función objetivo a lo largo de las iteraciones.

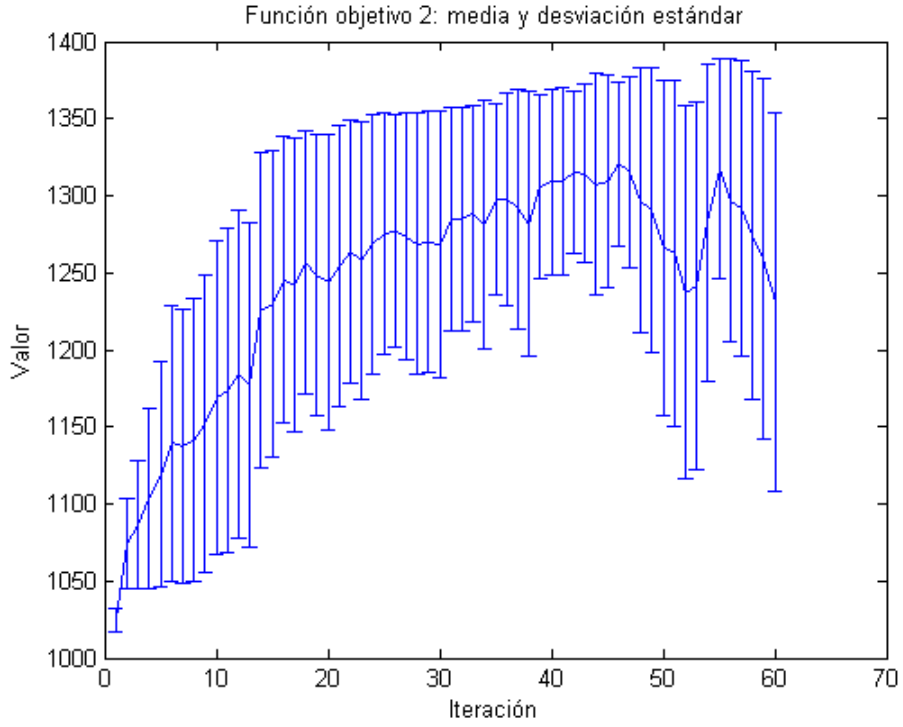


Figura 5: Evolución de la media y error (desviación estándar) del valor de la segunda función objetivo en la población.

Respecto del valor de la función objetivo asociada a calidad de las observaciones, en la figura 2 se observa que el valor mínimo fue disminuyendo, lo cual es clave para analizar un rango amplio de valores y así obtener distintas soluciones Pareto-eficientes. Por otro lado, el valor máximo también fue disminuyendo, pero suficientemente lento como para no empeorar la diversificación del algoritmo. Este punto es reforzado al analizar la figura 3, en que se presenta la evolución del valor promedio a lo largo de las iteraciones y que aún en las iteraciones donde este valor promedio no varió mucho, la diversidad de soluciones (ver las barras de error - desviación estándar) en términos de valor fue suficiente para que el algoritmo no presente convergencia prematura y por tanto poca exploración del espacio de soluciones factibles.

De forma análoga, en las figuras 4 y 5 vemos que en caso de la función que se deseaba maximizar, su mínimo y máximo fueron aumentando a lo largo de las iteraciones, y si bien el promedio no fue creciendo de forma monótona, la mantención de diversidad en la población también fue clave (observar el gráfico 5 entre la iteración 50 y 60).

Entre los 100 individuos de la población final, 69 de ellos resultaron no dominados, y varios comparten valor objetivo. El resultado final se resume en la figura 6. A modo de comparación, se observa en general la convergencia de la población al frente de Pareto propuesto, propiedad deseada que además señala que probablemente no se habrían observado mejoras considerables si el algoritmo hubiera continuado iterando. Sin embargo, hay que recordar que se desearía alcanzar mayor diversidad entre las soluciones Pareto-óptimas finales. La forma del frente es coherente, en el sentido que se esperaría un perfil similar cuando se busca minimizar el primer objetivo y maximizar el segundo.

Se hace necesario el uso de una población más grande y un mayor número de iteraciones para obtener una silueta más completa del frente de soluciones no dominadas. Para ello es crucial mejorar la rapidez de ejecución del algoritmo.

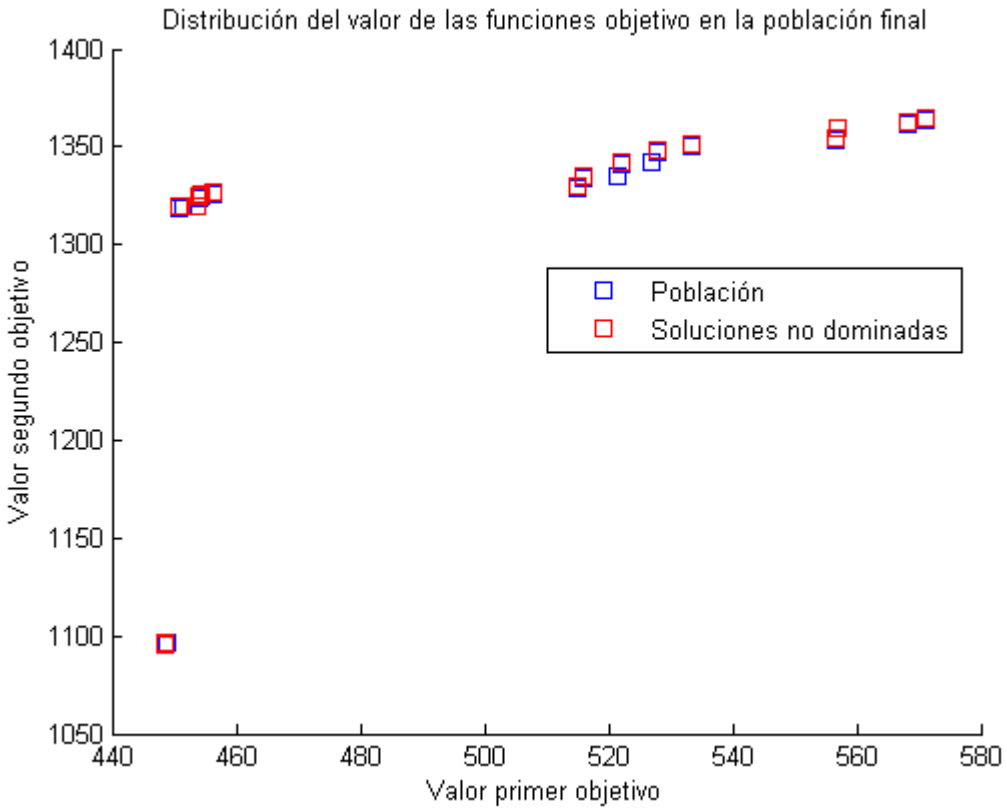


Figura 6: Aproximación del frente de Pareto obtenida por el algoritmo.

5. Conclusiones

En términos generales se logró conocer el potencial de Python, poder complementar con distintos paquetes vuelve al lenguaje una excelente herramienta a la hora de probar ideas concentrando esfuerzo en lo importante. Asimismo, se pudo conocer el área de optimización multiobjetivo, los conceptos básicos y algoritmos estándar que se utilizan en las aplicaciones. Además, se pudo dimensionar en un ejemplo la complejidad de los problemas reales que son enfrentados con métodos de optimización multiobjetivo.

La rutina de inicialización cumplió su propósito, pues logra generar planes de observaciones que cumplen las propiedades que se busca priorizar, es decir, bajos ángulos cenitales y visitas separadas para un mismo *target*. Respecto a los resultados del algoritmo genético: En ausencia de un frente de Pareto, en base a los gráficos y experimentos numéricos el algoritmo NSGA-II parece estar cumpliendo el propósito por el que se eligió. Son necesarias más pruebas numéricas para confirmarlo.

Para disminuir el tiempo de ejecución podría considerarse una discretización no sólo del cálculo de ángulos cenitales, sino también de los puntos factibles y de las distancias (en tiempo) entre pares de puntos. Lo anterior es crucial pues hasta el momento, al construir un plan factible se están calculando aproximadamente 800 veces las últimas 2 cantidades, por lo que un sacrificio en precisión a cambio de tiempo de cálculo podría marcar una diferencia.

Si bien se perdió mucho tiempo en la generación y revisión del código de la fase de reparación del algoritmo genético (por lo que no se alcanzó a discretizar más aspectos del algoritmo), los resultados de su aplicación son favorables, y probablemente necesarios para asegurar un mejor desempeño del algoritmo de optimización.

Se podría probar con otros operadores genéticos y funciones objetivo. En particular, para extender los operadores genéticos al *scheduling* de más de una noche el autor propone que se realicen mutaciones y *crossover* individualmente por noche. Además, se podría cambiar el operador de *crossover* por uno usado en problemas de permutaciones. Para más información ver el anexo.

Hay que recordar que el objetivo final del proceso de optimización multiobjetivo no es contar con el frente de soluciones no dominadas, sino tomar una decisión. En ese sentido, un posible criterio podría ser, entre todos los planes del frente, seleccionar el que tenga menos desfases (recordar que en la fase de reparación del algoritmo genético se permitió en algunos casos dejar un leve error en el tiempo entre observaciones). Una alternativa más general, al nivel del problema a 10 años, sería seleccionar la solución que cumpla más veces la restricción de visitas (visitar al menos 1000 veces cada *target*).

6. Anexo

En lo que sigue se presentan algunas alternativas que sería interesante evaluar en el trabajo futuro.

6.1. Comentarios finales y propuestas generales

Se podría calcular (de forma exhaustiva) el frente de Pareto para el scheduling de distintas noches individuales - o segmentadas por semanas - y así generar instancias *benchmark*. Para esto podría ser necesaria una capacidad computacional alta debido al crecimiento exponencial del número de combinaciones posibles, eventualmente el uso del supercomputador disponible en el CMM. La ventaja de contar con dicha información es que permitiría contar con instancias de referencia para testear los algoritmos que se generen en el futuro. Existen métricas para cuantificar la calidad de algoritmos de optimización multiobjetivo en términos de convergencia al frente de Pareto, como las mencionadas en [1]. El paquete DEAP cuenta con algunas rutinas implementadas, en que basta incluir el frente de Pareto y se procede a comparar con el frente obtenido por el algoritmo.

Respecto al tiempo de ejecución del código, hay un problema en el uso del paquete SCOOP que sólo permitió la paralelización de la fase de inicialización. Falta lograr una paralelización completa.

Para el caso de optimización dinámica, se propone evaluar el tiempo computacional asociado a cortar a partir de la noche de interrupción los planes de observación Pareto-óptimos de la población calculada anteriormente, y completar la solución usando planes factibles arbitrarios(no sesgados), para luego ejecutar el algoritmo genético sobre la nueva población. Si la rutina de generación de planes factibles sesgados lograra hacerse suficientemente rápida, también podría completarse la población con ese tipo de soluciones.

Otra opción es ocupar el algoritmo genético SPEA2 para optimización multiobjetivo. Dada la estructura del código, bastaría cambiar los operadores de selección por los adecuados.

6.2. Otras propuestas para funciones objetivo

Los algoritmos de optimización usados pueden obtener resultados diferentes si se utilizan otras funciones objetivo. En particular, a futuro sería interesante comparar el desempeño si se cambia el primer objetivo por maximizar

$$\sum_{i \in \mathcal{P}} h_i(t),$$

donde $z_i(t)$ corresponde a la altitud del *target* i en el tiempo t en que es observado. Notar que en realidad $h_i(t) = 90 - z_i(t)$, sin embargo, al estar minimizando la función objetivo usada en el trabajo los planes de observación que visiten menos objetos durante la noche tendrían mejor valor en la primera coordenada que el segundo. En el caso del algoritmo genético esto es compensado con la rutina que deja los planes lo más extensos posibles (es decir, no se deja tiempo muerto durante o al final de la noche, se considera que el telescopio siempre está o moviéndose o realizando una observación). Sin embargo, en el futuro podrían analizarse otras posibilidades y perderse esta característica.

Otra idea interesante puede ser maximizar el mínimo de las altitudes a lo largo del plan (en lugar de la suma), o minimizar el máximo de los ángulos cenitales.

6.3. Otras propuestas para el operadores genéticos

El operador de mutación usado puede ser muy local, debido a lo largos que son los planes de observación (con alrededor de 800 puntos por noche en el caso sin cambio de filtros). Por lo tanto, se propone como

alternativa el operador de mutación que con cierta probabilidad baja (por ejemplo, 0.02 o $\frac{1}{L}$, donde L es el largo del *schedule*). De esta manera se pueden presentar mutaciones de más de un punto a la vez.

En el caso de *crossover*, dado que a nivel de una noche tanto en el problema estudiado como en el general los planes de observación debieran verse como una sub-permutación del conjunto de *targets*, se propone evaluar el uso de operadores de *crossover* específicos para problemas de permutaciones (por ejemplo, el Problema del Vendedor Viajero).

Referencias

[1] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). *A fast and elitist multiobjective genetic algorithm: NSGA-II.* Evolutionary Computation, IEEE Transactions on, 6(2), 182-197.

[2] De Rainville, F. M., Fortin, F. A., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). *DEAP: A Python Framework for Evolutionary Algorithms.*

[3] Eiben, A. E., & Smith, J. E. (2008). *Introduction to evolutionary computing* (Natural Computing Series).

[4] Förster, F., López, N., Maza, J., Kubánek, P., & Pignata, G. (2009). *Scheduling in targeted transient surveys and a new telescope for CHASE.* Advances in Astronomy, 2010.

[5] Kubanek P., *Genetic algorithm for robotic telescope scheduling*, M.S. thesis, Universidad de Granada, Granada, Spain, 2007.

[6] Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

[7] The Large Synoptic Survey Telescope - http://www.lsst.org/. Consultado el 28-12-2015.