

## Aspectos técnicos de los robots co-trabajadores con operadores humanos

Raúl Suárez, Jan Rosell

Recibido: 8 de Marzo de 2019 / Aceptado: 10 de Abril de 2019

### Resumen

La robótica colaborativa, en la que los robots comparten el espacio de trabajo con los humanos e interactúan con ellos, toma un papel cada vez más relevante. En este contexto, el artículo presenta desarrollos realizados en diferentes aspectos técnicos necesarios para que estos robots funcionen adecuadamente y cumplan las expectativas en cuanto a su integración. En particular, se enfatizan las capacidades de planificación autónoma de tareas y movimientos, de manipulación diestra, de manipulación bimanual y de interacción con los humanos. Todos los desarrollos presentados han sido validados experimentalmente, y en el artículo se incluyen ejemplos ilustrativos de los experimentos realizados.

### Palabras clave

Robótica, manipuladores móviles, robots co-trabajadores, cooperación humano-robot.

## 1. Introducción

El campo de aplicación de la robótica crece constantemente a medida que los avances tecnológicos van permitiendo nuevas prestaciones. Entre las nuevas aplicaciones destaca, por su potencial trascendencia tanto desde el punto de vista productivo como desde el social, aquellas en las que los robots trabajan conjuntamente con los seres humanos, dando origen al concepto de “robot co-trabajador”. En estas aplicaciones los robots requieren características particulares. Por un lado, han de tener cierta autonomía y capacidad de decisión, ya que su función no es la de realizar tareas de forma repetitiva en el sentido clásico, sino que deberán adaptarse con cierta celeridad a condiciones cambiantes, sobretudo generadas por los operarios con los que comparten el entorno de actuación. Esta adaptación implica una capacidad de interacción con los operarios que condiciona los movimientos y forma de actuar de los robots, a fin de evitar posibles daños a la integridad física de los humanos, pero evidentemente manteniendo la eficiencia de sus acciones. Por otra parte, para que los robots co-trabajadores sean realmente útiles es necesario, además, que tengan un nivel de destreza suficiente que les permita ejecutar diferentes acciones usualmente requeridas en las actividades humanas.

En este contexto, se presentan aquí resultados de trabajos orientados al desarrollo de robots autónomos diestros como co-trabajadores con operadores humanos. Estos resultados pretenden incrementar las prestaciones de los robots co-tra-

bajadores para acercarlos a su implantación definitiva en nuestra sociedad, y facilitar también su aceptación por parte de los humanos.

Concretamente, se plantean avances en temas tan relevantes como:

- El aumento de la capacidad del robot para gestionar autónomamente las tareas y los movimientos específicos para llevarla a cabo, representando el conocimiento mediante ontologías.
- El incremento de la capacidad de manipulación diestra y bimanual, monitorizando las acciones para prevenir potenciales fallos.
- La mejora en la interacción con el operario humano, en referencia a movimientos reactivos, de cooperación y de transferencia de objetos.

Todos los desarrollos teóricos aquí presentados se han probado y validado experimentalmente mediante sistemas específicamente preparados para tal fin. Por otra parte, los problemas mencionados se abordan con la intención de aportar soluciones generales que sean válidas tanto en la robótica industrial como en la de servicios, previendo tanto la influencia puramente industrial de los resultados como la componente social de los mismos.

Tras esta introducción, el resto del artículo está estructurado de la siguiente manera. La sección 2 presenta la motivación y el alcance del trabajo. A continuación se presentan las diferentes capacidades que se requieren en los robots para conseguir los objetivos marcados: capacidad de gestión autónoma de tareas y movimientos (Sección 3), capacidad de manipulación diestra (Sección 4), capacidad de manipulación bimanual (Sección 5) y capacidad de interacción con



Raúl Suárez  
[raul.suarez@upc.edu](mailto:raul.suarez@upc.edu)

Jan Rosell  
[jan.rosell@upc.edu](mailto:jan.rosell@upc.edu)

Institut d'Organització i Control de Sistemes Industrials  
(IOC - Divisió de Robòtica)  
Universitat Politècnica de Catalunya (UPC)

los operadores humanos (Sección 6). Finalmente, la Sección 7 presenta las conclusiones.

## 2. Motivación

Aunque la introducción de robots avanza de manera imparable en distintos ámbitos, la disponibilidad de robots humanoides con capacidades “casi humanas” y que realicen tareas avanzadas en el trabajo y en el hogar no es inmediata. Sin embargo, la existencia de robots manipuladores móviles con capacidades de manipulación diestra que puedan asistir al ser humano de manera versátil en distintas tareas simples es ya una realidad, por ejemplo, como transportadores logísticos o adaptaciones de manipuladores clásicos, y algunos fabricantes ya comercializan modelos que cumplen las normativas de seguridad requeridas para compartir espacio de trabajo con los seres humanos. En cualquier caso, lograr una adecuada efectividad, depende de mejoras de hardware y de software que permitan un grado elevado de adaptabilidad. El uso de robots como compañeros de trabajo es un salto evolutivo en la robótica industrial. La disponibilidad de robots que puedan realizar distintas tareas, autónomamente o en cooperación, y compartir de forma segura el espacio con los humanos puede permitir incrementar la producción y los puestos de trabajo. Los robots co-trabajadores necesitan la interacción y supervisión humanas, pueden considerarse como herramientas avanzadas y es posible establecer con ellos una relación simbiótica, permitiendo realizar más tareas en menos tiempo, con mayor calidad y de forma más comfortable. La propuesta de introducir en el entorno de trabajo robots co-trabajadores tiene un carácter disruptivo que puede tener efectos importantes sobre el modelo productivo actual, incluyendo el comportamiento y la convivencia laboral. Por ello, en los desarrollos en este campo hay que tener en cuenta la importancia de los comportamientos y percepciones sociales que esta tecnología pueda generar, ampliando sus ventajas y minimizando los riesgos asociados para que estos comportamientos propicien el proceso de innovación, que debe conllevar un aumento de la productividad y de la calidad, así como una mejora de las condiciones de trabajo. En este sentido, deben implementarse sistemas en los que los robots co-trabajadores den soporte y se integren en las tareas de los operarios de manera fluida, natural, mínimamente invasiva y segura. Ahora bien, para que el concepto de robots co-trabajadores se consolide, tanto en el ambiente industrial como en el de servicios, aún se deben realizar avances científicos y tecnológicos en determinados temas afines. Por ejemplo, aunque ya existen propuestas y ensayos experimentales, la manipulación diestra y la cooperación bimanual en su sentido más riguroso aún son temas de trabajo en centros de investigación, así como la capacidad de cooperación con el operador humano en cuanto a planificación simultánea de movimientos y tareas para una adecuada interacción, desde el punto de seguridad, de eficiencia, y también de confort para el humano. Por lo tanto, resulta razonable y pertinente trabajar en estos aspectos.

Así, entre las motivaciones más relevantes para abordar los temas que se describen en las secciones siguientes podemos mencionar:

- Los robots co-trabajadores se presentan actualmente como una de las herramientas de alto nivel más versátiles y útiles;
- Se dispone en el mercado de robots manipuladores móviles de prestaciones mecánicas aceptables, pero que tienen fuertes limitaciones en cuanto a destreza y capacidad de interacción con el operario humano;
- Entre las causas de estas limitaciones están las carencias en la capacidad de manipulación diestra, por ejemplo explotando inadecuadamente el uso de información táctil, y en la capacidad de cooperación segura y eficiente con el humano, por ejemplo no planificando adecuadamente las tareas y movimientos en concordancia con los del humano.

## 3. Capacidad de gestión autónoma de tareas y movimientos

Muchos retos en robótica, como las tareas de manipulación en entornos desordenados, requiere de la planificación a nivel de tarea y a nivel de movimientos. La planificación a nivel de movimientos se centra en la búsqueda de caminos libres de colisión entre dos configuraciones del robot, y usualmente se realiza en el espacio de configuraciones mediante métodos basados en muestreo (García et al., 2017) y pueden requerir realimentación tanto de configuración como de fuerza cuando se prevé contacto (Suárez et al. 1995). Diferentes variantes permiten la inclusión de distintos tipos de restricciones, e incluso permiten la interacción con objetos móviles del entorno, como los planificadores basados en la física que consideran las interacciones dinámicas (robot-objeto y objeto-objeto) durante la planificación. La planificación a nivel de tarea, por el contrario, se centra en buscar una secuencia de acciones que permita la ejecución de una determinada tarea. Usualmente se modela como un sistema de transición de estados y puede resolverse, por ejemplo, usando métodos de satisfacción de restricciones o métodos heurísticos de búsqueda en el espacio de estados, como el método Fast Forward (FF) (Hoffmann y Nebel, 2001).

Las tareas a abordar pueden tener diferentes grados de dificultad, tanto a nivel simbólico (dimensión del espacio de estados) como geométrico (conectividad de las zonas libres del espacio de configuraciones), así como de interacción entre ambos niveles. Los estados y acciones a nivel simbólico deben implementarse a nivel geométrico, siendo las acciones ejecutables solo si las implementaciones geométricas satisfacen las precondiciones de la acción y si la acción es geoméricamente factible. Por ello, es necesaria una combinación eficiente entre los niveles de planificación para hacer que el proceso sea eficiente.

Asimismo, el uso del conocimiento y de técnicas de razonamiento pueden ayudar a las capacidades de planificación a ambos niveles, dando más autonomía a los robots. El conocimiento puede modelarse usando ontologías, que lo representan mediante conceptos y relaciones. Las ontologías pueden implementarse mediante el Web Ontology Language (OWL) (McGuinness et al., 2004), haciéndolo una base de datos accesible a través de la red, que puede ser usada para el razonamiento mediante predicados escritos en lenguaje Prolog.

Unas de las tareas de manipulación de gran interés en el marco de los robots co-trabajadores es la manipulación de objetos usando un robot bi-brazo. En estas tareas se debe hacer frente a la existencia de objetos que bloquean los movimientos del robot, requiriendo una correcta selección de las configuraciones de prensión y de regiones para depositar los objetos manipulados, y garantizando que las acciones sean realizables. Asimismo, en función del tipo de problema a resolver, el orden en que se resuelven los objetivos parciales debe ser analizado cuidadosamente, ya que es posible la generación de espacios de estados muy grandes y soluciones que impliquen mover varias veces los mismos objetos para conseguir el objetivo final deseado.

En esta sección se describe un marco basado en ontologías para modelar tareas de manipulación y un planificador simultáneo de tareas y de movimientos basado en el conocimiento llamado k-TMP. Este método combina un planificador de tareas heurístico Fast Forward con un planificador de movimientos basado en la física (que permite la realización de tareas de empuje de objetos sin sujetarlos) y con ontologías. Tanto el planificador de movimientos como el conocimiento modelado con las ontologías se usan en procesos de razonamiento para determinar la factibilidad de las acciones y guiar al planificador de tareas en la búsqueda del plan de acciones. El planificador de movimientos basado en la física también se usa en acciones de prensión en entornos desordenados cuando las posiciones finales de los otros objetos no son relevantes, ya que este planificador permite de manera natural apartar objetos que estorban en el camino hacia la prensión de un objeto predeterminado.

### 3.1. Planificación combinada de tareas y movimientos

Considérese un robot bi-brazo capaz de manipular objetos (de distintas formas y tamaños) en una mesa, es decir de sujetarlos (posiblemente con diferentes agarres) y de dejarlos (en diferentes sitios como bandejas, estantes, o zonas intermedias), así como de desplazarlos empujándolos. En este escenario se pueden plantear distintas tareas, que van desde la simple prensión de un objeto determinado hasta la reordenación de varios objetos. Para encontrar la secuencia de acciones factible que permita solventar la tarea deben superarse distintos retos, como la detección de los objetos bloqueantes,

la selección de los agarres que son cinemáticamente posibles y libres de colisiones tanto al sujetar como al dejar un objeto, la determinación de dónde dejar los objetos teniendo en cuenta la alcanzabilidad y las posibles limitaciones de espacio, y la determinación de caminos libres de colisión del robot, moviéndose sin carga o en el proceso de manipulación de un objeto.

Para abordar estos problemas se propone un marco para combinar los niveles de planificación basado en:

- La definición de ontologías de manipulación: La formalización del conocimiento de manipulación que incluye información de los objetos, robots, sensores y acciones, el entorno y el contexto, así como un mecanismo de inferencia para razonar sobre este conocimiento.
- La planificación para la prensión: Un planificador de movimientos basado en la física capaz de buscar trayectorias que incluyan la interacción con objetos móviles, permitiendo apartar aquellos que interfieren en el camino al objeto a sujetar, permitiendo prensiones simples y eficientes.
- Planificación de la manipulación: Un planificador de tareas basado en la búsqueda heurística en el espacio de estados, en el que se guía la búsqueda en base a procesos de razonamiento geométrico que determinan la factibilidad y aplicabilidad de las acciones.

En las siguientes subsecciones se detallan estos puntos.

### 3.2. Ontologías para la manipulación

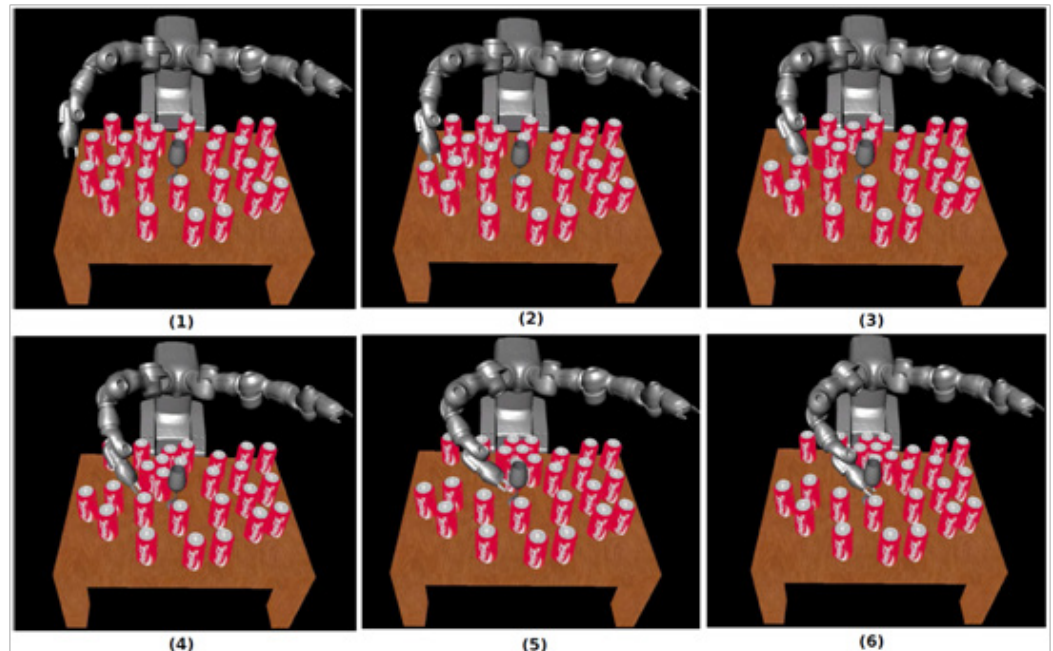
El marco propuesto para la representación del conocimiento necesario para tareas de manipulación usa un planificador de movimientos basado en la física, que automáticamente permite la categorización de los objetos en diferentes tipos en base a las restricciones de los objetos y de la tarea (Diab et al., 2017). La propuesta usa vocabulario común dado por el Ontologies for Robotics and Automation Working Group (ORA WG), auspiciado por IEEE Robotics & Automation Society (Olszewska et al., 2017), basado en una jerarquía de ontologías de tres niveles: metaontology (para representar información general, como el concepto de objeto rígido), ontology schema (para representar conocimiento de un dominio específico, como un entorno de oficina) y ontology instance (para almacenar la información de objetos particulares). Estos niveles se subdividen en clases para representar el conocimiento relativo a: Características (propiedades de los objetos relevantes para la manipulación), Objeto (objetos físicos y sus partes y agrupaciones), Actor (robots y sus componentes), Espacio (espacio de trabajo, relaciones topológicas), Contexto (conocimiento relativo a la situación temporal y espacial) y Acción (algoritmos de planificación).

### 3.3. Planificación de movimientos de prensión

Para la planificación de movimientos de prensión se presenta un planificador basado en la física que permite abordar la búsqueda de trayectorias robustas del robot en entornos donde puede haber interacciones con los objetos y donde hay incertidumbre en las posiciones de los objetos y en los parámetros de interacción (Muhayyuddin et al., 2018). La propuesta se basa en el planificador KPIECE (Sucan y Kavraki, 2012), un planificador basado en muestreo que explora el espacio de configuraciones mediante un árbol guiado mediante una proyección en un subespacio discretizado donde se evalúa la exploración (el proceso de expansión escoge con mayor probabilidad aquellas ramas cuyas celdas asoci-

adas tienen un mayor índice de calidad). Se trabaja en dos fases: muestreo y cálculo de certeza. Una vez seleccionada la rama del árbol que debe crecer, se seleccionan aleatoriamente direcciones de crecimiento generando potenciales extensiones. En la fase de evaluación de la certeza se considera incertidumbre (en el posicionamiento de los objetos, en las direcciones de crecimiento y en los parámetros de interacción) y se estima la tasa de éxito que la ejecución de cada extensión reportaría, seleccionando la mejor. La certeza de la nueva rama se usa para actualizar el índice de la celda del subespacio asociado, promoviendo un sesgo hacia zonas con más certeza. Este método se ha probado con diferentes robots (Kuka LWR y ABB YuMi) y con diferentes escenas con muchos objetos (Figura 1).

**Figura 1** Ejemplo de tarea de prensión en un entorno desordenado.



### 3.4. Planificación de la manipulación

Para la planificación de tareas de manipulación se propone un planificador simultáneo de tareas y movimientos basado en el planificador de tareas Fast Forward (FF), que realiza una búsqueda heurística en el espacio de estados, en el que se incorpora información geométrica en el cálculo de la heurística y se incluyen llamadas al planificador de movimientos en las acciones que llevan a los estados explorados (Akbari et al., 2018). El enfoque propuesto tiene tres partes principales: el cálculo de la heurística, la búsqueda en el espacio de estados y el proceso de selección de las acciones.

La heurística usada en el planificador FF se estima como la distancia al estado final, calculada como el número de acciones de un plan obtenido mediante una versión relajada del Grafo de Planificación (Blum y Furst, 1997), llamada RPG. La modificación propuesta incluye un análisis geométrico de las acciones del plan relajado basado en averiguar, por ejem-

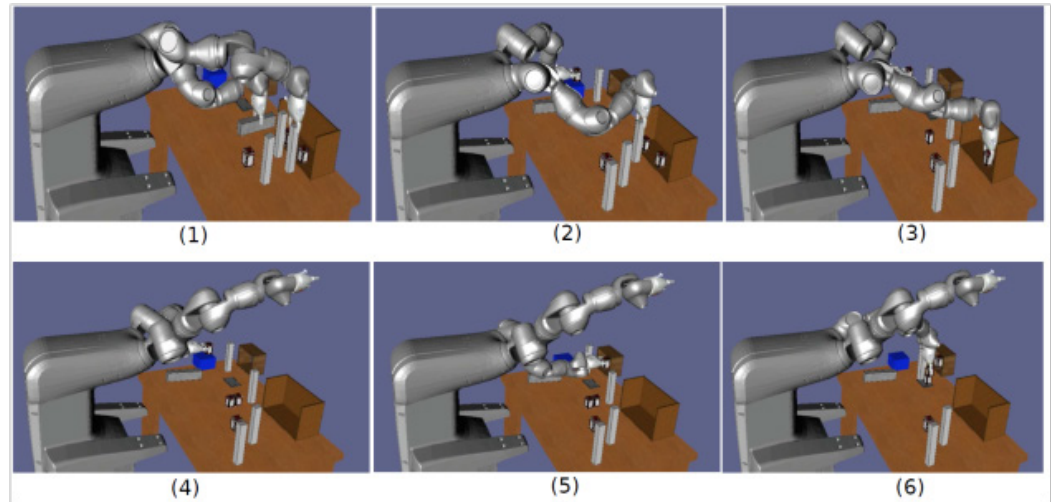
plo, si existen soluciones de la cinemática inversa para las configuraciones de agarre, si el robot colisiona o no con el entorno en dichas configuraciones, o si existe espacio suficiente para poder dejar los objetos. Si todas las acciones del plan relajado son factibles, entonces el valor de la heurística es el número de acciones de dicho plan. En caso contrario, se identifica la razón y se actualiza el estado (por ejemplo, etiquetando un objeto como obstáculo) y se recalcula un nuevo plan relajado. Este proceso de razonamiento permite una búsqueda informada en términos geométricos.

La búsqueda en el espacio de estados se hace, igual que el FF estándar, mediante el algoritmo Enhanced Hill Climbing (EHC) y alternativamente con el algoritmo Best First Search (BFS). Desde cada estado el módulo selecciona el estado sucesor que tiene un valor de la heurística menor.

El proceso de selección de la acción busca un camino para una acción usando el planificador de movimientos adecuado (un planificador geométrico para encontrar caminos libres de colisión y un planificado basado en la física (physics based) para movimientos de empuje). Si la planificación de movimientos falla, el estado se actualiza y recomienza la búsqueda. En caso de éxito se almacena el camino juntamente con la acción.

La propuesta se ha validado con éxito en tareas de sobremesa ejecutadas con el robot bi-brazo YuMi (Figura 2). La tarea requiere quitar diferentes objetos que bloquean el paso, seleccionar con tino el brazo que realiza cada acción, seleccionar adecuadamente las configuraciones de agarre de las piezas, y ordenar adecuadamente la consecución de objetivos.

**Figura 2** Ejemplo de una tarea de manipulación ejecutada con el robot YuMi.

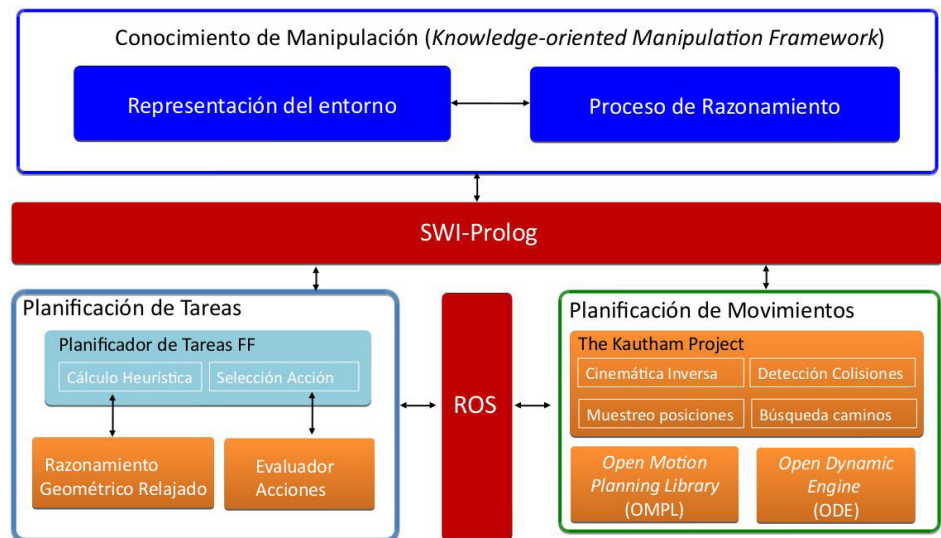


**3.5. Implementación**

El marco de planificación propuesto se ha implementado en tres partes (Figura 3): planificación de la tarea, planificación de movimientos y procesado del conocimiento. La planificación de la tarea se ha implementado modificando una versión de planificador FF en C++ para incluir la información geométrica y el procesado del conocimiento. La planificación de movimientos se ha hecho usando The Kautham

Project (Rosell et al., 2014), que permite la planificación de movimientos basándose en la librería Open Motion Planning Library (Sucan et al., 2012). El procesado del conocimiento se ha realizado en Prolog, y la generación de las ontologías mediante el editor Protege. Todas las comunicaciones se han hecho usando el Robotic Operating System (ROS) (Quigley et al., 2009).

**Figura 3** Implementación: componentes y comunicaciones.



## 4. Capacidad de manipulación diestra

Un robot co-trabajador debe tener en sí mismo capacidades avanzadas de manipulación, lo que incluye aspectos tales como el uso coordinado de dos brazos para llevar a cabo una tarea (sea a cadena cinemática abierta o cerrada cuando dos brazos sujetan un único objeto) o la capacidad de manipular objetos de forma diestra. En este contexto, el concepto de manipulación diestra se refiere a la manipulación de objetos dentro del espacio de trabajo de la mano usando solamente los dedos (in-hand manipulation).

En entornos de trabajo reales generalmente el modelo del objeto a manipular solo es parcialmente conocido, o incluso totalmente desconocido, lo que significa que no se dispone de su modelo para precalcular los movimientos de manipulación. En estos casos, una solución útil consiste en utilizar sensores táctiles que ayudan al reconocimiento del objeto en los puntos de contacto y por lo tanto a construir su modelo geométrico.

Se describe a continuación un enfoque propuesto para realizar manipulación diestra con dos dedos (Montaño y Suárez, 2015, 2018), cuyas principales asunciones son:

- La mano robótica tiene un sistema de sensores táctiles que permite obtener información del punto de contacto y de la fuerza ejercida sobre el objeto manipulado, y no se dispone de otros sensores externos como, por ejemplo, realimentación visual. Si se dispone de información sobre la cinemática de la mano y de la posición de sus articulaciones (información propioceptiva).
- Se utilizan dos dedos para manipular el objeto emulando la sujeción mediante los dedos pulgar e índice de la mano humana.
- No se tiene información previa sobre la forma del objeto, pero se asume que es un sólido rígido.
- No se conoce el coeficiente de fricción, pero se asume que está por encima de un valor crítico prefijado en función del material de la yema del dedo, que es conocido.

- La mano robótica utilizada tiene un sistema de control de posición de las articulaciones de los dedos, por lo que el enfoque propuesto cierra un bucle de control de más alto nivel que genera comandos de posición para este sistema de control.

La manipulación diestra persigue normalmente tres objetivos, individual o conjuntamente:

- Desde el punto de vista de la mano, la optimización de su configuración, es decir, que la configuración sea lo más próxima posible a una configuración predeterminada. Frecuentemente, se considera como configuración predeterminada la definida por la posición media de las articulaciones a fin de dar, potencialmente, máxima capacidad de movimiento. Para esta optimización se minimiza el siguiente índice (Liégeois 1977):

$$I_{hc} = \sum_{i=1}^2 \sum_{j=1}^{n_i} \left( \frac{q_{ij} - q_{oij}}{q_{maxij} - q_{minij}} \right)^2 \quad [1]$$

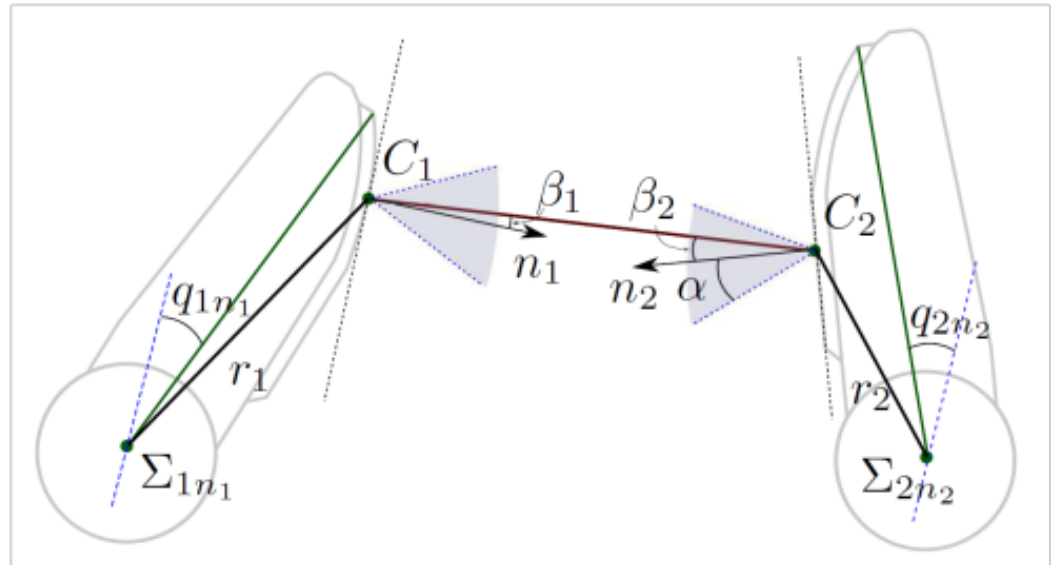
donde  $q_{ij}$  es la posición de la articulación  $j$  del dedo  $i$ ,  $q_{oij}$  su valor deseado (predefinido), y  $q_{maxij}$  y  $q_{minij}$  sus límites máximo y mínimo.

- Desde el punto de vista de la presión (relación objeto-mano), la optimización de la calidad de la presión, es decir, la búsqueda de presiones seguras en las que la mano pueda resistir mayores fuerzas externas aplicadas al objeto. Existen diferentes medidas de calidad para evaluar la presión de un objeto (Roa et al., 2008; Roa y Suárez, 2014). En este trabajo, para esta optimización se minimiza el siguiente índice:

$$I_{gq} = \frac{1}{2} \sum_{i=1}^2 |\beta_i| \quad [2]$$

donde  $\beta_i$ ,  $i \in \{1,2\}$  es el ángulo entre el segmento que une los dos puntos de contacto y las normales  $n_i$ , en cada uno de ellos (ver Figura 4). Cuanto menores son los ángulos  $\beta_i$  mayor es el margen de seguridad entre la fuerza aplicada y el borde del cono de fricción.

**Figura 4** Extremos de los dedos donde se puede apreciar las normales a las superficies de contacto y los ángulos  $\beta_i$  que indican cuán lejos están las fuerzas aplicadas de los extremos del cono de fricción.



- Desde el punto de vista del objeto, la optimización de su configuración, es decir, la búsqueda de una posición y orientación que satisfagan los requisitos de una tarea dada. En este trabajo los dedos se encargarán de orientar adecuadamente el objeto, mientras que el brazo es el encargado de corregir la posición. Para esta optimización se minimiza el siguiente índice:

$$I_{oe} = \left| \frac{\gamma_d - \gamma}{\gamma_d - \gamma_i} \right| \quad [3]$$

donde  $\gamma_d$  es la orientación deseada del objeto,  $\gamma$  es su orientación actual y  $\gamma_i$  es la orientación inicial.

El procedimiento general usado en cada iteración  $k$  para calcular las posiciones de los dedos deseadas en el paso  $k+1$  puede resumirse como (véanse las Figuras 5 y 6):

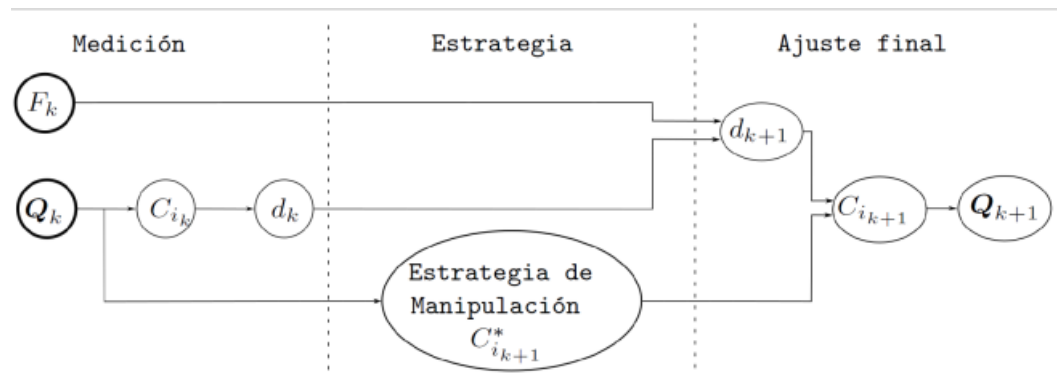
Etapa de medición. Se mide la posición de las articulaciones de los dedos de ambas manos incluyendo la posición

de los respectivos puntos de contacto sobre los sensores táctiles,  $Q_k$ , así como la fuerza aplicada por los dedos sobre el objeto,  $F_k$ . Con esta información es posible calcular la posición absoluta de los puntos de contacto en el espacio  $C_{ik}$  y la distancia entre ellos  $d_k$ .

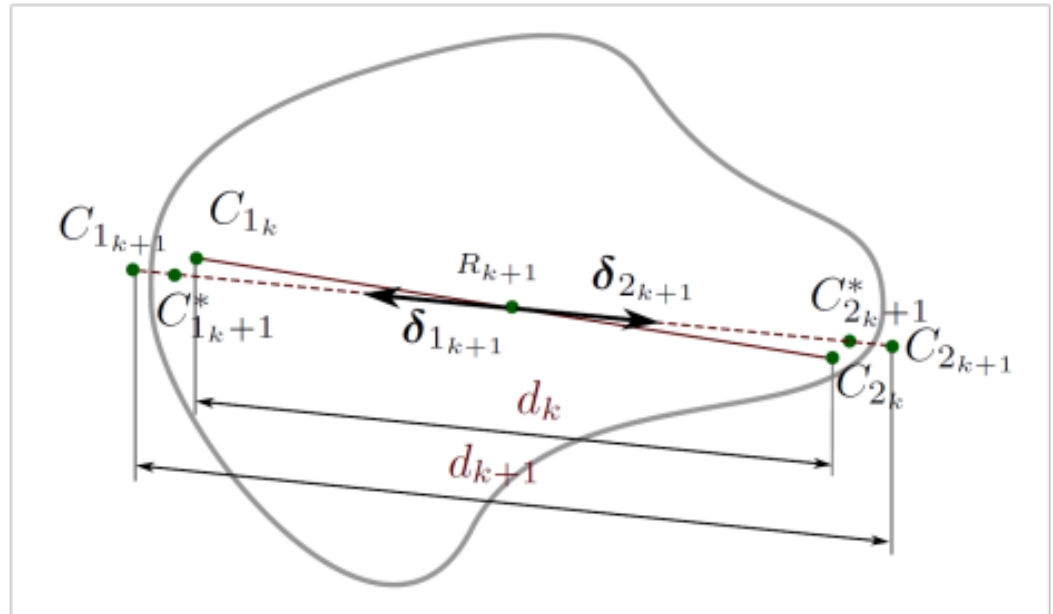
Aplicación de la estrategia deseada. Conocido el vector  $Q_k$ , se calculan unos nuevos puntos de contacto,  $C_{ik+1}^*$ , de forma que al mover los dedos hacia estas posiciones se produzca el efecto deseado (Montaño y Suárez, 2018).

Ajuste final. Conocida la fuerza actual  $F_k$  y la distancia actual entre los contactos en cada dedo  $d_k$ , se calcula la nueva distancia deseada  $d_{k+1}$  (el cálculo se detalla a continuación) apretando o relajando la presión sobre el objeto según corresponda, y, finalmente, la distancia entre los puntos  $C_{ik+1}^*$  se ajusta al valor  $d_{k+1}$  moviéndolos de forma simétrica respecto a su punto medio para obtener así los puntos finales  $C_{ik}$  que realmente servirán de consigna para el movimiento de los dedos al pasar de la iteración  $k$  a la  $k+1$ .

**Figura 5** Etapas en el cálculo de la configuración de los dedos  $Q_{k+1}$  en cada iteración



**Figura 6** Ejemplo del cálculo de  $C_{ik+1}$  usando  $C_{ik+1}^*$  ajustando la distancia  $d_k$  a  $d_{k+1}$  cuando la fuerza de contacto  $F_k$  es mayor que la cota máxima prefijada  $F_{max}$ .



Dado que la forma del objeto desconocida, durante su manipulación las fuerzas pueden variar saliendo del rango de seguridad deseado, para evitar este efecto la distancia  $d$  se ajusta en cada iteración en función de la diferencia entre el valor de la fuerza de presión actual  $F_k$  y la fuerza deseada  $F_d$ , i.e. el error  $e=F_k-F_d$ . Así,

$$d_{k+1} = d_k + \Delta d \quad [4]$$

donde

$$\Delta d = \begin{cases} f_1(e) & \text{si } e \leq 0 \\ f_2(e) & \text{si } e > 0 \end{cases} \quad [5]$$

siendo  $f_1(e)$  y  $f_2(e)$  dos funciones arbitrariamente definidas por el usuario. En este trabajo se ha usado  $f_1(e)=2\lambda(e+e^2)$  y  $f_2(e) = \lambda e$ , siendo  $\lambda$  un valor constante predefinido. La razón por la que  $f_1(e)$ , tiene una mayor ganancia según aumenta  $|e|$ , es que una potencial caída del objeto ( $F_k \rightarrow 0$ ) se considera más una situación más crítica que una potencial aplicación de fuerzas de presión superiores a las deseadas ( $F_k \gg F_d$ ), pero esto puede ajustarse fácilmente dependiendo de la aplicación.

Finalmente, debe destacarse que el proceso de manipulación iterativo puede finalizar por distintas causas:

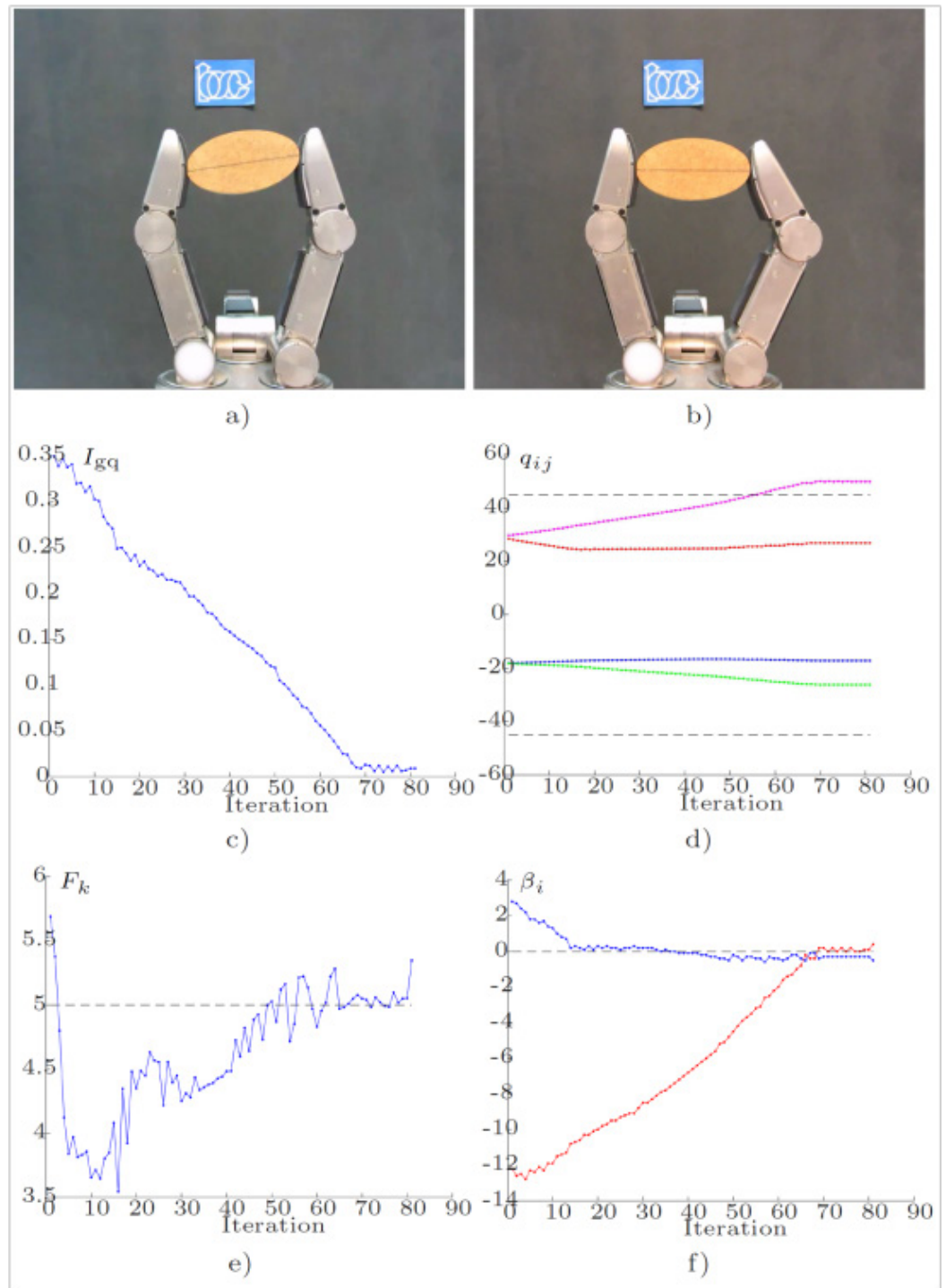
- El índice de calidad utilizado alcanza su valor óptimo.

- El valor máximo actual del índice de calidad utilizado no ha mejorado pasado un número prefijado de iteraciones (no es necesario que el índice mejore monóticamente, puede decrecer u oscilar localmente).
- La presión esperada con los nuevos puntos de contacto calculados no satisface las restricciones de fricción y el objeto se puede caer.
- Los nuevos puntos de contacto calculados no pertenecen al espacio de trabajo de la mano y por lo tanto no son alcanzables.

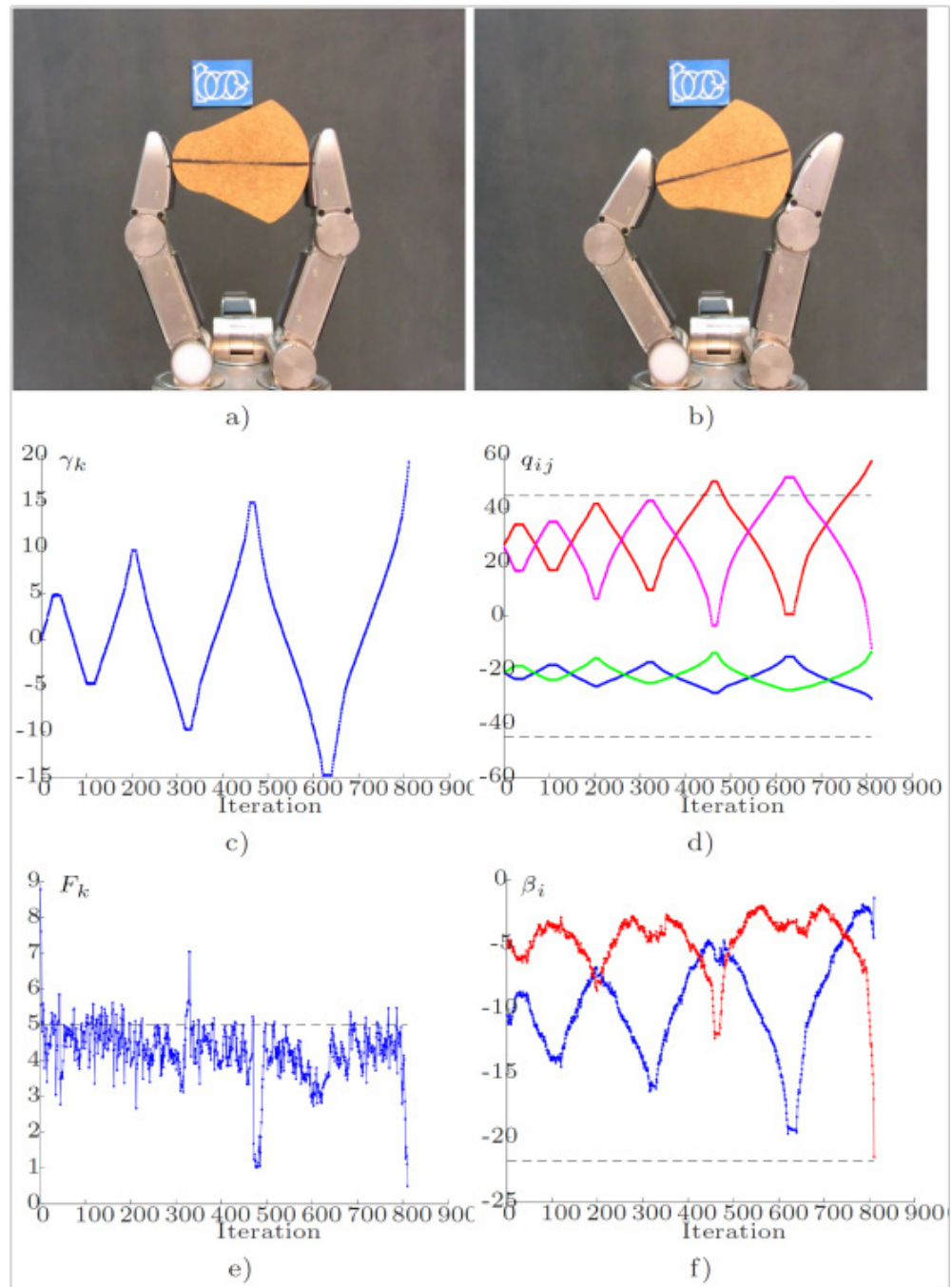
En las Figuras 7 y 8 se ilustran dos ejemplos de manipulación diestra. En el primer caso se optimiza la calidad de la presión  $I_{gq}$  y la manipulación finaliza después de 5.460 s y 81 iteraciones debido a que  $I_{gq}$  ya no mejora su valor. En el segundo ejemplo se varía la orientación del objeto en ambos sentidos siguiendo unas consignas  $\gamma_d$  igual a 5, -5, 10, -10, 15, -15 y 20 grados; el cambio de consigna se realiza cada vez que se alcanza una condición de finalización, en el caso de las primeras 6 consignas  $I_{oe}$  alcanzó el valor óptimo, i.e.  $\gamma_k=\gamma_d$  y en la última consigna el procedimiento se detuvo cuando en el próximo paso se esperaba que el ángulo  $\beta_2$  excediera el límite del cono de fricción antes de llegar a  $\gamma_k=20$  grados, lo que implica un riesgo de deslizamiento del objeto en los dedos con su consiguiente caída.



**Figura 7** Optimización de la calidad de la prensión: a) Prensión inicial. b) Prensión final. c) Evolución del índice  $I_{gq}$ . d) Evolución de los valores de las articulaciones (en grados),  $q_{11}$  en azul,  $q_{12}$  en rojo,  $q_{21}$  en verde,  $q_{22}$  en magenta. e) Fuerza  $F_k$  (en Newtons, la línea a trazos indica  $F_d$ ). f) Ángulos  $\beta_i$  (en grados),  $\beta_1$  en azul y  $\beta_2$  en rojo (la línea a trazos indica el valor óptimo  $\beta_i=0$ ).



**Figura 8** Optimización de la orientación del objeto: a) Prensión inicial. b) Prensión final. c) Evolución de la orientación  $\gamma_k$ . d) Evolución de los valores de las articulaciones (en grados),  $q_{11}$  en azul,  $q_{12}$  en rojo,  $q_{21}$  en verde,  $q_{22}$  en magenta. e) Fuerza  $F_k$  (en Newtons, la línea a trazos indica  $F_d$ ). f) Ángulos  $\beta_i$  (en grados),  $\beta_1$  en azul y  $\beta_2$  en rojo (la línea a trazos indica el valor óptimo  $\beta_i=0$ ).



Aparte de la manipulación diestra autónoma de objetos, el enfoque descrito se ha usado también en tareas de teleoperación de una mano robótica (Montaño y Suárez, 2016a, 2017). En esta aplicación se tiene un sistema de control compartido entre el operador y el sistema robótico remoto, en el que el operador solo envía comandos de alto nivel indicando en qué sentido se debe rotar el objeto, y el sistema remoto usa un grafo de estados y la información táctil y cinemática de la mano para controlar localmente las fuerzas de prensión y los movimientos de los dedos, generando así manipulaciones robustas y estables que evitan caídas del objeto independientemente de los posibles retardos que pudieran existir en el canal de comunicación.

### 5. Capacidad de manipulación bimanual

La manipulación bimanual puede entenderse de dos modos, uno en que las dos manos son parte de una cadena cinemática cerrada, por ejemplo, cuando sujetan un mismo objeto, y otro en que cada mano es el elemento terminal de una cadena cinemática independiente pero trabajando coordinadamente, por ejemplo, cuando las dos manos mueven diferentes objetos cooperando tras un mismo fin.

#### 5.1. Sujeción bimanual de objetos voluminosos

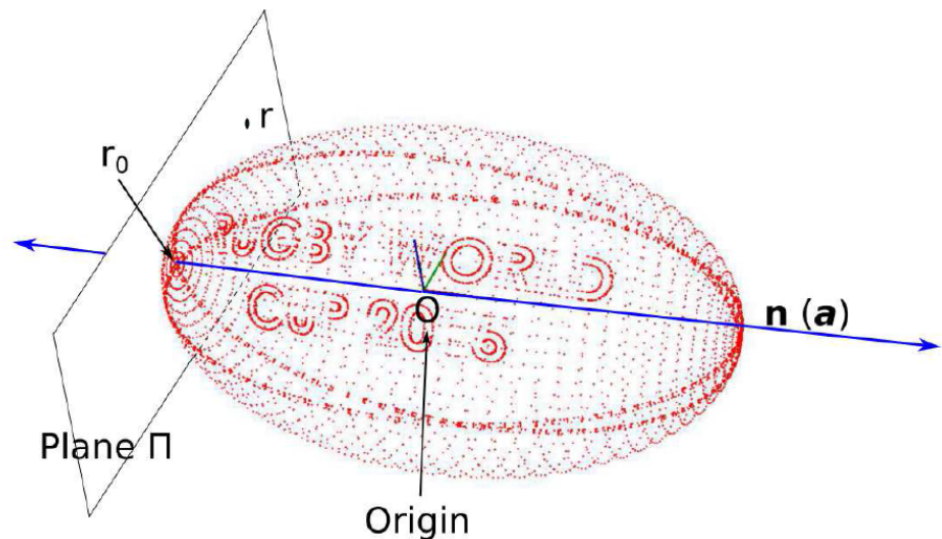
Uno de los principales problemas abordados en manipulación bimanual es la sujeción robusta de objetos voluminosos que no pueden sujetarse con una sola mano. Se muestra

aquí un ejemplo de planificador de prensiones bimanuales basado en una segmentación del objeto (Rojas-de-Silva y Suárez, 2016; Rodríguez et al., 2016).

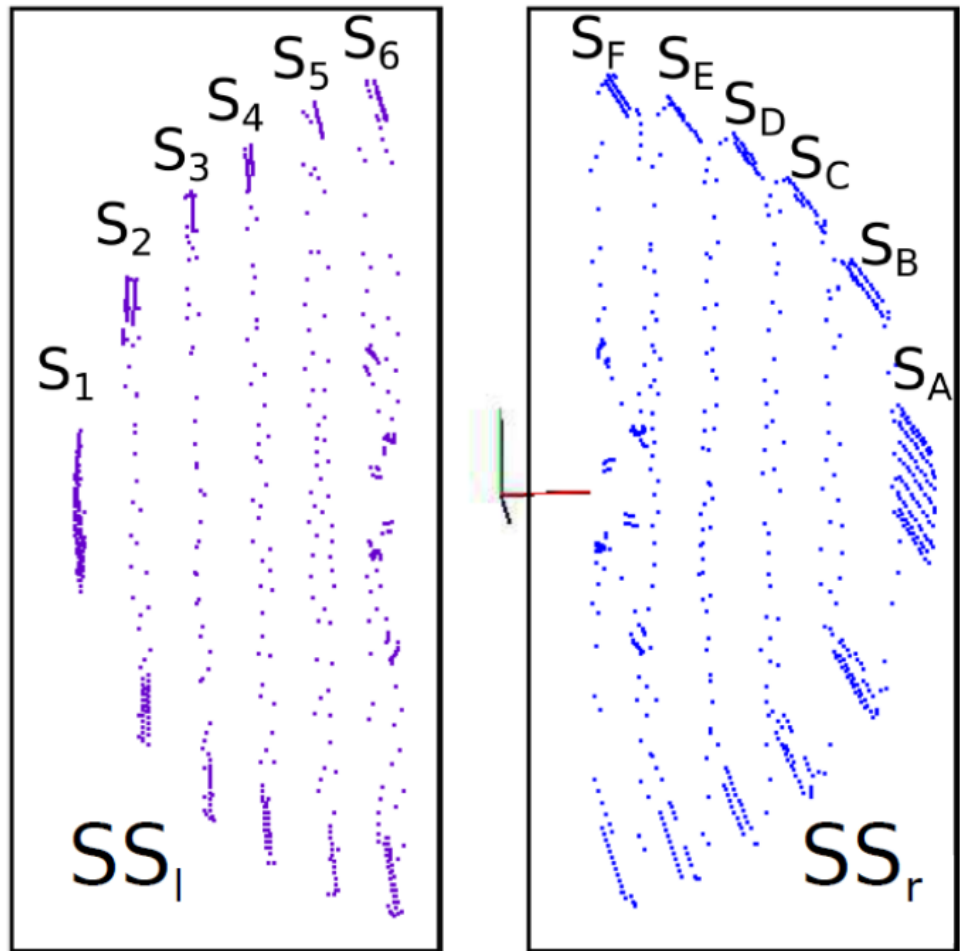
El objeto se representa mediante una nube de puntos 3D cuya información asociada incluye la dirección normal al objeto en cada punto. En el cálculo de la prensión se consideran contactos puntuales con fricción y la implementación realizada puede utilizar tres o cuatro dedos de cada mano, lo que da redundancia y mayor seguridad a la prensión.

Se describe a continuación el procedimiento de planificación particularizado para el uso de tres dedos de cada mano. El planificador busca los tres puntos de contacto alcanzables por cada mano dividiendo la nube de puntos que representa la superficie del objeto en dos conjuntos de rebanadas ortogonales a uno de los ejes de inercia del objeto. Se comienza por el eje de inercia que forma el menor ángulo con respecto al segmento definido por las bases de los brazos, es decir el segmento que va de "hombro a hombro", y si con este no se encuentra una solución se prueba con otro eje de inercia elegido con el mismo criterio. Para definir cada rebanada se usan planos ortogonales al eje de inercia usado y a una distancia predeterminada entre ellos, siendo el primero de estos planos tangente al objeto (Figura 9) y el último a una distancia del primero equivalente a la máxima distancia alcanzable por los dedos sobre un objeto cualquiera sin que éste penetre en palma de la mano (Figura 10).

**Figura 9** Objeto representado por una nube de puntos y el primer plano ortogonal a un eje de inercia de los usados para dividir el objeto en rebanadas.



**Figura 10** Conjunto de rebanadas resultantes del objeto mostrado en la Figura 9. Cada rectángulo incluye las rebanadas de un extremo del objeto



Una vez definidas las rebanadas, se selecciona iterativamente una tripleta de puntos de una rebanada de cada extremo del objeto. Las rebanadas se van explorando desde el extremo hacia el centro del objeto avanzando secuencialmente de un lado y del otro, asegurando de este modo, dentro de lo posible, una cierta coherencia en las posiciones de las manos al sujetar el objeto. A fin de eliminar tripletas de puntos claramente no alcanzables por los dedos, se usan los siguientes tres filtros:

- El área del triángulo formado por los tres puntos de la tripleta debe ser menor que el área del triángulo formado por los puntos centrales de las yemas de los tres dedos usados de cada mano cuando estos están extendidos;
- La distancia entre los puntos debe ser menor que la máxima distancia posible entre las yemas de los dedos de la mano usada;
- Un índice Q que indica que tan cerca está el triángulo definido por los tres puntos de ser equilátero debe estar por encima de un umbral dado. Q se calcula como

$$Q = \frac{(b+c-a)(c+a-b)(a+b-c)}{abc} \quad [6]$$

siendo a, b y c las longitudes de los lados del triángulo. Q es igual a 1 para triángulos equiláteros e igual a 0 para triángulos degenerados de área nula.

Para cada par de tripletas de puntos se verifica si haciendo contacto en cada uno de los seis puntos se puede obtener una presión que satisfaga la condición de equilibrio de fuerzas (force-closure, FC), es decir, que pueda resistir perturbaciones de fuerzas externas en cualquier dirección, y si tiene una calidad superior a un umbral deseado prefijado. La calidad de la presión está dada por el módulo de la máxima perturbación (fuerza y/o par) que la presión puede resistir en cualquier dirección (Ferrari y Canny, 1992).

Una vez encontrado un par de tripletas que permita una presión FC con la calidad deseada, se realiza un análisis de alcanzabilidad para ver si los puntos seleccionados son alcanzables por el sistema bi-brazo. La complejidad de este paso es elevada ya que se debería probar la alcanzabilidad para cada combinación posible de asignación de los seis puntos de contacto a cada uno de los seis dedos que se van a usar, y en cada caso se ha de calcular la cinemática inversa del sistema robótico completo, es decir, los dos brazos con sus respectivas manos. Para reducir la complejidad se usa una estrategia que simplifica el proceso de asignación de los

puntos a los dedos. Los puntos se agrupan en dos conjuntos dependiendo de la dirección de la normal (exterior) a la superficie del objeto: cuando la normal apunta en dirección a la posición del robot los puntos son candidatos a ser contactados por el pulgar, y, en caso contrario por el resto de

los dedos (ver Figura 11). Cuando el conjunto de puntos es cinemáticamente alcanzable por los dedos correspondientes de cada mano se procede a planificar los movimientos de los brazos y manos para llegar a ellos usando planificadores de movimientos basados en muestreo (García et al., 2017).

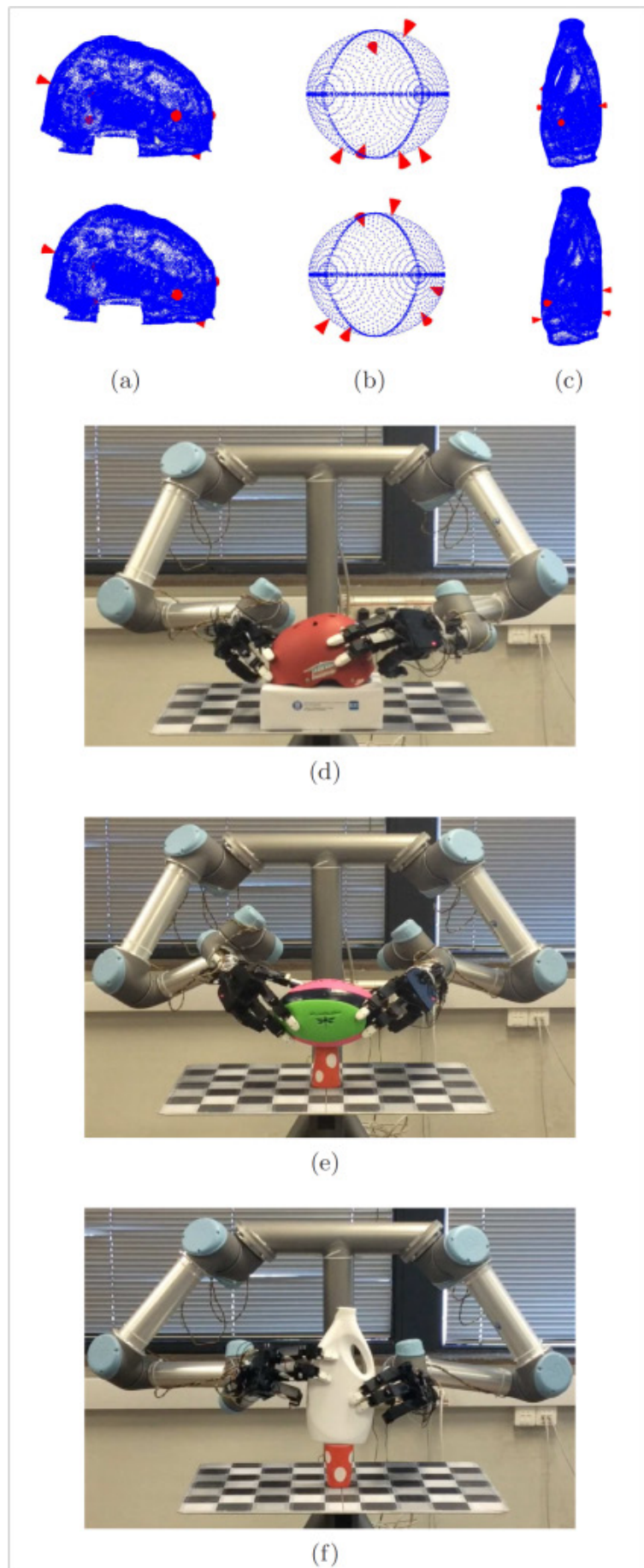
**Figura 11** Puntos candidatos a ser contactados en la prensión, en rojo los puntos asignados al pulgar, en azul los asignados al resto de los dedos.



El planificador de prensiones bimanuales ha sido aplicado a distintos tipos de objetos. En la Figura 12 se muestran tres ejemplos de prensiones válidas de un casco, de un balón de rugby y de una botella de detergente, mostrándose los puntos

de contacto obtenidos sobre el modelo del objeto, así como la ejecución de la prensión con un sistema robótico bi-brazo real.

**Figura 12** Ejemplos de prensiones bimanuales: a)-c) Ejemplos de puntos de contacto alcanzables por el sistema bi-braza sobre los modelos de los objetos; d)-f) Prensiones con el sistema robótico real.



La continuación de este enfoque está orientada a optimizar las fuerzas que deben aplicarse en las articulaciones de los dedos para optimizar a su vez las fuerzas resultantes sobre el objeto siguiendo diferentes criterios, como, por ejemplo, hacer que los dedos hagan las fuerzas mínimas necesarias para la sujeción, hacer que el objeto esté sometido a las menores fuerzas internas posibles, o intentar evitar equilibrar las fuerzas que hace cada mano al sujetar el objeto (Rojas-de Silva y Suárez, 2017).

## 5.2. Cooperación bimanual en la manipulación independiente de objetos

Otra forma de cooperación bimanual es la manipulación coordinada en la que ambos brazos trabajan en la misma tarea pero sin interacción física entre ellos. Dentro de este tipo de colaboración se han realizado desarrollos con el objetivo de sujetar dos objetos determinados, uno con cada mano, considerando que el acceso a estos objetos puede ser obstaculizado por otros objetos que deberán ser oportunamente apartados usando ambas manos (Rodríguez et al., 2014; Rodríguez y Suárez, 2016). Los objetos deseados pueden sujetarse con cualquiera de las dos manos o pueden ser pre-asignados a cada una de ellas, y lo mismo sucede con los obstáculos, que pueden ser apartados con cualquiera de las dos manos siguiendo distintas estrategias de optimización.

En estas aplicaciones se han utilizado variantes de planificadores de movimientos basados en muestreo, como PRM o RRT (García et al., 2017). En estas variantes, cuando se calculan las trayectorias de los brazos hasta cada objeto no se descartan las configuraciones en las que se producen colisiones con obstáculos que puedan ser quitados, sino que se acepta el camino, pero marcando que solo es válido si se quitan tales obstáculos. De este modo, el planificador de movimientos se utiliza de forma recursiva, comenzando por buscar caminos hasta los objetos de interés, y siguiendo con la búsqueda de caminos para quitar los obstáculos encontrados, y así sucesivamente. De esta forma se genera un grafo de precedencias  $G$ , con dos nodos raíces que representan los objetos de interés y otros que, agregados recursivamente, representan los obstáculos se han de quitar para poder acceder (y mover) cada objeto con cada brazo. El grafo  $G$  es del tipo AND/OR (Stoffel et al., 1995), cada nodo representa un

objeto  $O_j$  ( $j = 1, 2, \dots, n$ ) y cada arco que sale de él lo liga a otro nodo que representa un obstáculo existente en el camino  $P_{ij}$  encontrado para que el brazo  $R_i$  ( $i = 1, 2$ ) acceda a  $O_j$ . La Figura 13 muestra un ejemplo de un grafo  $G$  que se detalla más adelante.

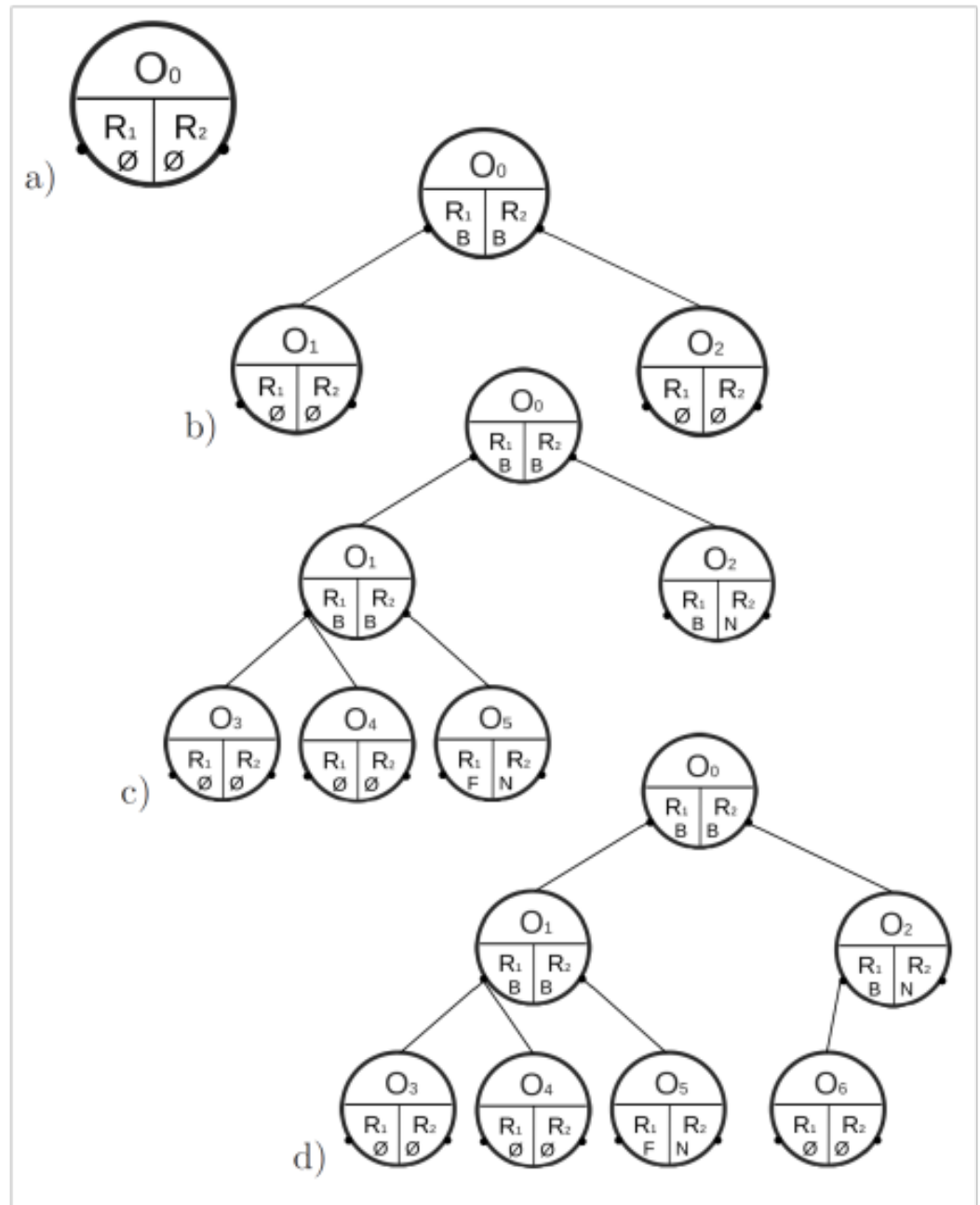
El conjunto  $O_{ij}$  agrupa a todos los obstáculos existentes en  $P_{ij}$ .

Cada nodo  $O_j$  tiene las siguientes propiedades:

- Un identificador (id) que indica el objeto representado  $O_j$ .
- Un identificador (root id) que indica cuál es el nodo padre de  $O_j$ .
- Un coste (root cost) para cada nodo raíz del grafo que representa el número de nodos (objetos) que, desde  $O_j$ , deben quitarse para alcanzar el correspondiente nodo raíz.
- Un coste local (local cost) que indica el número objetos que con un brazo dado  $R_i$  se han de quitar para acceder a  $O_j$ .
- El estado (state) de  $O_j$ , que indica para cada brazo  $R_i$ :
  - Empty ( $\emptyset$ ), cuando no se ha calculado aun un camino  $P_{ij}$  para alcanzar  $O_j$  con  $R_i$ .
  - Null (N), cuando  $R_i$  no puede alcanzar  $O_j$  (i.e.  $O_j$  está fuera del alcance cinemático de  $R_i$ ).
  - Free (F), cuando existe un camino  $P_{ij}$  para que  $R_i$  alcance  $O_j$  sin obstáculos ( $SO_{ij} = \emptyset$ ).
  - Blocked (B), cuando ya se tiene un camino  $P_{ij}$  para que  $R_i$  alcance  $O_j$  pero con obstáculos ( $SO_{ij} \neq \emptyset$ ).

La Figura 13 muestra un ejemplo de un grafo  $G$  para un único objeto de interés (i.e. con un único nodo raíz). Del análisis del grafo  $G$  se obtiene una secuencia de acciones para cada brazo que permite acceder a los objetos de interés siguiendo distintos criterios, como, por ejemplo, quitar el menor número posible de obstáculos para acceder a los objetos de interés.

**Figura 13** Ejemplo de grafo de precedencias G: (a) nodo raíz representando el objeto de interés  $O_0$ ; (b) usar  $R_1$  para manipular  $O_0$  produce  $SO_{1,0}=\{O_1\}$  y usar  $R_2$  produce  $SO_{2,0}=\{O_2\}$ , luego  $O_1$  y  $O_2$  se agregan como nodos hijos de  $O_0$  con arcos para  $R_1$  y  $R_2$  actuando como arcos OR; (c) quitar  $O_1$  con  $R_1$  produce  $SO_{1,1}=\{O_3, O_4\}$  luego  $O_3$  y  $O_4$  se agregan como nodos hijos de  $O_1$  con dos arcos AND para  $R_1$ , mientras que quitar  $O_1$  con  $R_2$  produce  $SO_{1,1}=\{O_5\}$  por lo que  $O_5$  se agrega como hijo de  $O_1$ , actuando en este caso como arco OR para  $R_2$ ; (d) en la otra rama, quitar  $O_2$  con  $R_1$  produce  $SO_{1,2}=\{O_6\}$  luego  $O_6$  se agrega como hijo de  $O_2$  con un arco OR para  $R_1$ , y cuando se intenta quitar  $O_2$  con  $R_2$  no se encuentra un camino válido (por ejemplo debido a que no hay solución cinemática para que el brazo  $R_2$  sujete  $O_2$ ) y por lo tanto no se generan nuevos nodos hijos de  $O_2$  para  $R_2$ . Este procedimiento se ejecuta secuencialmente hasta que los nodos terminales de G representan objetos que pueden ser manipulados sin apartar ningún obstáculo y por lo tanto se puede buscar una secuencia de acciones para apartar los obstáculos hasta llegar a  $O_0$ .

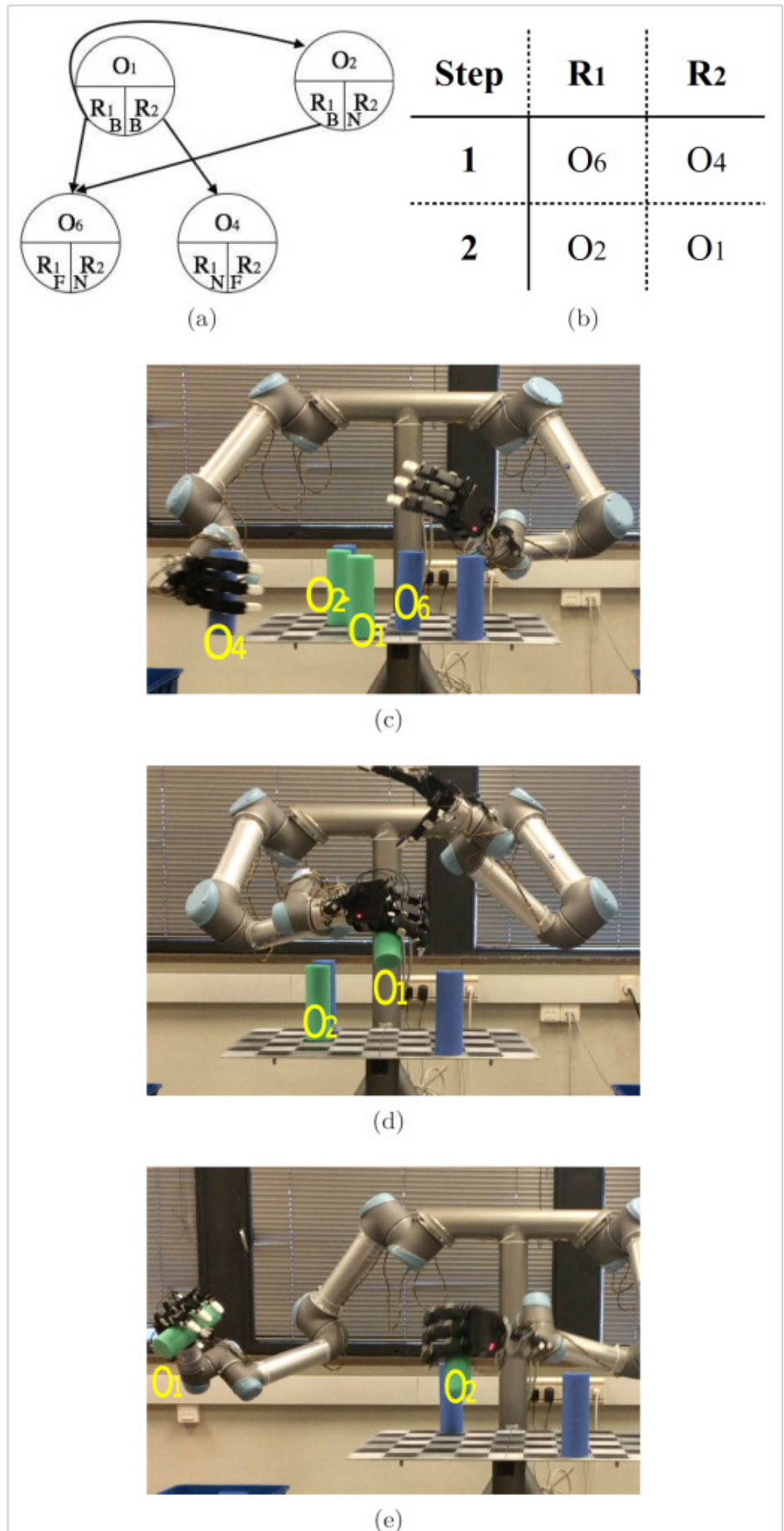


Los caminos geométricos obtenidos para cada brazo con este procedimiento pueden ejecutarse paralelizando algunas acciones, pero como no se ha realizado ninguna coordinación temporal podrían producirse colisiones entre los brazos. Para evitar estas colisiones, el movimiento de los brazos siguiendo los caminos geométricos obtenidos se coordina en línea (Montaña y Suárez, 2016b), priorizando las tareas de acuerdo con su precedencia en el grafo y ajustando la evolución temporal de los movimientos de los brazos sin cambiar sus rutas geométricas.

La Figura 14 ilustra el procedimiento. Cada mano debe sujetar uno de los dos objetos de interés, los cilindros verdes  $O_1$  y  $O_2$ , mientras que los cilindros azules  $O_{i,i \neq 1,2}$  son obstáculos amovibles. La Figura 14a muestra el grafo de precedencias obtenido, de su análisis se obtuvo como resultado que, en una primera ronda de movimientos, el robot  $R_1$  es elegido para quitar el obstáculo  $O_6$  y el robot  $R_2$  para quitar el obstáculo  $O_4$ , para que posteriormente  $R_1$  sujete el objeto deseado  $O_1$  y  $R_2$  el objeto deseado  $O_2$ . Las Figuras 14(b-d) muestran algunas fotos de la ejecución real del experimento.



**Figura 14** Ejemplo de cooperación bi-brazo, los objetos de interés son los cilindros verdes  $O_1$  y  $O_2$ : a) Grafo de precedencias; b) secuencia de acciones, el brazo  $R_1$  quita  $O_6$  y el  $R_2$  quita  $O_2$  en paralelo, luego  $R_1$  sujeta  $O_1$  y  $R_2$  sujeta  $O_4$ ; c), d) y e) instantáneas de la ejecución real, con  $R_1$  yendo a sujetar  $O_6$  y  $R_2$  sujetando  $O_4$ ,  $R_2$  moviendo  $O_1$  y  $R_1$  esperando para sujetar  $O_2$ , y  $R_1$  moviendo  $O_2$  con  $R_2$  ya en su pose final.



Actualmente se trabaja en nuevos modelos que faciliten la búsqueda de soluciones óptimas de forma sistemática, y considerando también la posibilidad de realizar la transferencia de un objeto de una mano a otra y de colaborar con el operario. En esta línea ya se han propuesto medidas de calidad que indiquen cuan buena es una prensión de cara a la posterior transferencia del objeto (Soler et al., 2018).

## 6. Capacidad de interacción con operarios humanos

Es necesario que los robots co-trabajadores tengan la capacidad de interacción con los operarios humanos para llevar a cabo tareas colaborativas. Por un lado, el sistema debe ser capaz de planificar tareas en las que el operario humano es uno de los actores en su ejecución y el robot co-trabajador el otro. Por otro, al compartir el mismo espacio de trabajo, es indispensable que no haya colisiones entre el robot y los operarios humanos y que éstos se puedan mover con total confianza y seguridad. Estos retos se detallan en las siguientes subsecciones.

### 6.1. Planificación contingente de tareas

Respecto al primer reto, se ha incorporado el operario como actor en el planificador de tareas, para realizar ciertas tareas que el robot difícilmente puede realizar, para darle información al robot sobre el estado del entorno, y para realizar acciones de transferencia de objetos al espacio alcanzable del robot. Las acciones realizadas por el operario pueden no tener un resultado único, por lo que se ha usado la variante del planificador Fast Forward, llamada Contingent Fast Forward (Brafman y Hoffmann, 2004), que genera un árbol de planes que contiene todas las posibles variantes en función de los diferentes resultados de las acciones realizadas por el operario.

### 6.2. Manipulación con movimientos de apariencia humana

Respecto al segundo reto, es necesario que el robot esté dotado de la capacidad de realizar movimientos con apariencia humana, para que el operario pueda predecir intuitivamente sus movimientos y pueda moverse con tranquilidad a su lado. Asimismo, el robot debe poder detectar la presencia y del operario humano y tener definidas estrategias reactivas que eviten colisiones. Esta fase está actualmente en desarrollo, y se prevé el uso de un sistema de cámaras que puedan monitorizar continuamente los movimientos del operario humano, y, acorde a estos, adaptar ligeramente de forma dinámica las trayectorias planificadas o, de ser necesario, replanificarlas.

Para planificar movimientos con apariencia humana, se plantea primero una transformación entre las articulaciones de un operador humano y las de un robot antropomorfo (aun cuando el robot pueda tener una estructura cinemática algo diferente), y se usa para trasladar al robot los movimientos que realiza un humano al ejecutar distintas tareas. A continuación, se estudian las relaciones entre las variaciones de las articulaciones del robot cuando realiza estos movimientos, tanto en lo relativo a posición como a velocidades (denominadas sinergias de orden cero y uno respectivamente). Estas sinergias permiten definir un subespacio donde realizar la planificación de movimientos, obteniéndose movimientos con apariencia humana y, además, de manera eficiente al reducirse la dimensión del espacio de planificación (García et al., 2017, 2018).

## 7. Conclusiones

La tendencia de la robótica industrial se dirige hacia la robótica colaborativa, en la que los robots comparten el espacio de trabajo con los operadores humanos e interactúan con ellos, tomando un papel de co-trabajadores activos. En este artículo se han presentado diferentes aspectos técnicos que este reto conlleva. Por un lado, aspectos que permiten una convivencia segura en el mismo entorno, como la ejecución de movimientos con apariencia humana. Por otro, aspectos que permiten a los robots ser más autónomos, como las capacidades avanzadas de manipulación, incluyendo manipulación diestra y cooperación bimanual, así como la capacidad de planificación simultánea de movimientos y tareas. Estos desarrollos deben permitir a los robots trabajar en entornos menos estructurados, facilitando su integración en la pequeña y mediana empresa, permitiendo un aumento de la producción y mejorando la calidad de los puestos de trabajo.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno Español mediante el proyecto DPI2016-80077-R.

Los autores agradecen la colaboración de Aliakbar Akbari, Mohammed Diab, Néstor García, Andrés Montaña, Muhayyuddin, Carlos Rodríguez, Abiud Rojas-de-Silva y Leopold Palomo.

## 8. Referencias

Akbari, A., Laggrifoul, F., y Rosell, J. (2018). A sampling-based tree planner for systems with complex dynamics. *Autonomous Robots*.

- Blum, A.L. y Furst, M.L. (1997). Fast planning through planning graph analysis. *Artificial intelligence*, 90(1), 281–300.
- Brafman, R. y Hoffmann, J. (2004). Conformant planning via heuristic forward search: A new approach. In S. Koenig, S. Zilberstein, y J. Koehler (eds.), *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS-04)*, 355–364. Whistler, Canada.
- Diab, M., Muhayyuddin, Akbari, A., y Rosell, J. (2017). An ontology framework for physics-based manipulation planning. In *ROBOT 2017: Third Iberian Robotics Conference. Advances in Intelligent Systems and Computing*, volume 693, 452–464. Springer, Cham.
- Ferrari, C. y Canny, J. (1992). Planning optimal grasps. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2290–2295. IEEE Comput. Soc. Press.
- García, N., Rosell, J., y Suárez, R. (2017). Aplicación de algoritmos RRT en la planificación de movimientos óptimos en robótica. In *Proc. of the 12th Metaheuristics International Conference (MIC'17) - Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'17)*, 953 – 962.
- García, N., Rosell, J., y Suárez, R. (2018). Motion planning by demonstration with human-likeness evaluation for dual-arm robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–10. doi: 10.1109/TSMC.2017.2756856.
- García, N., Suárez, R., y Rosell, J. (2017). Task-dependent synergies for motion planning of an anthropomorphic dual-arm system. *IEEE Transactions on Robotics*, 33(3), 756–764. doi: 10.1109/TRO.2017.2676131.
- Hoffmann, J. y Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 253–302.
- Liégeois, A. (1977). Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12), 868–871.
- McGuinness, D., Van Harmelen, F., et al. (2004). Owl web ontology language overview. *W3C recommendation*, 10(10), 2004.
- Montaño, A. y Suárez, R. (2015). Unknown object manipulation based on tactile information. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5642–5647.
- Montaño, A. y Suárez, R. (2016a). Commanding the object orientation using dexterous manipulation. In *Advances in Intelligent Systems and Computing*, volume 418, 69–79. Springer Verlag.
- Montaño, A. y Suárez, R. (2016b). Coordination of several robots based on temporal synchronization. *Robotics and Computer-Integrated Manufacturing*, 42, 73–85.
- Montaño, A. y Suárez, R. (2017). Robust dexterous tele-manipulation following object-orientation commands. *Industrial Robot: the international journal of robotics research and application*, Vol. 44, Issue 5, 2017, pp. 648–657 (ISSN: 0143-991X)..
- Montaño, A. y Suárez, R. (2018). Manipulation of unknown objects to improve the grasp quality using tactile information. *Sensors*, 18(5). doi:10.3390/s18051412.
- Muhayyuddin, Moll, M., Kavragi, L., y Rosell, J. (2018). Randomized physics-based motion planning for grasping in cluttered and uncertain environments. *IEEE Robotics and Automation Letters*, 3(2), 712–719. doi: 10.1109/LRA.2017.2783445.
- Olszewska, J.I., Barreto, M., Bermejo-Alonso, J., Carbonera, J., Chibani, A., Fiorini, S., Goncalves, P., Habib, M., Khamis, A., Olivares, A., de Freitas, E.P., Prestes, E., Ragavan, S.V., Redfield, S., Sanz, R., Spencer, B., y Li, H. (2017). Ontology for autonomous robotics. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., y Ng, A.Y. (2009). ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, volume 3, 5.
- Roa, M.A., Suárez, R., y Cornellà, J. (2008). Medidas de calidad para la prensión de objetos. *Revista iberoamericana de Automática e Informática Industrial*, 5(1), 66–82.
- Roa, M.A. y Suárez, R. (2014). Grasp quality measures: review and performance. *Autonomous Robots*, 38(1), 65–88.
- Rodríguez, C., Rojas-de-Silva, A., y Suárez, R. (2016). Dual-arm framework for cooperative applications. In *IEEE 21st Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*.
- Rodríguez, C. y Suárez, R. (2016). Combining motion planning and task planning for a dual-arm system. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2016)*, 4238–4243.

- Rodríguez, C., Montaña, A., y Suárez, R. (2014). Planning manipulation movements of a dual-arm system considering obstacle removing. *Robotics and Autonomous Systems*, 12(12), 1816–1826.
- Rojas-de-Silva, A. y Suárez, R. (2016). Grasping bulky objects with two anthropomorphic hands. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 877–884.
- Rojas-de-Silva, A. y Suárez, R. (2017). Contact Force Computation for Bimanual Grasps. In *Proceedings of 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2017)*, 1–6. IEEE Comput. Soc. Press.
- Rosell, J., Pérez, A., Aliakbar, A., Muhayyuddin, Palomo, L., y García, N. (2014). The Kautham Project: A teaching and research tool for robot motion planning. In *IEEE Int. Conf. on Emerging Technologies and Factory Automation*.
- Soler, F., Rojas-de Silva, A., y Suárez, R. (2018). *Grasp Quality Measures for Transferring Objects*, 28–39. Springer International Publishing.
- Stoffel, D., Kunz, W., y Gerber, S. (1995). *AND/OR graphs*. Technical Report - Max-Planck Institut für Informatik, Germany.
- Suárez R., Basañez, L., y Rosell J. (1995). Using Configuration and Force Sensing in Assembly Task Planning and Execution. In *IEEE Int. Symp. on Assembly and Task Planning, ISATP'95*, (IEEE Pres., ISBN 0-8186-6995-0), Pittsburgh, Pennsylvania, EEUU, 1995, pp. 273-279.
- Sucan, I. y Kavraki, L.E. (2012). A sampling-based tree planner for systems with complex dynamics. *IEEE Transactions on Robotics*, 28(1), 116–131.
- Sucan, I., Moll, M., y Kavraki, L.E. (2012). The open motion planning library. *Robotics & Automation Magazine, IEEE*, 19(4), 72–82.

## Robust Car Sequencing Problem: Description, models and metrics

Joaquín Bautista-Valhondo

Recibido: 8 de Marzo de 2019 / Aceptado: 10 de Abril de 2019

### Summary

The concept of partial demand uncertainty is introduced from the car sequencing problem (CSP) and incorporates special vehicle fleets in a demand plan. After establishing the working hypotheses with fleets, a mixed integer linear programming model, robust car sequencing problem (r-CSP), is proposed to satisfy the maximum number of CSP restrictions. Subsequently, multi-sequence production and metrics to evaluate its robustness are defined. The r-CSP considers various demand scenarios and functions to measure the excess of optional requirements in production programs. These functions are valid as objectives in optimization problems and as multi-sequence production robustness metrics.

### Keywords

Assembly lines, Sequencing, Car Sequencing Problem, Vehicle Fleets, Robust Optimization, Production Mix, Demand uncertainty.

## 1. Introduction

In the manufacturing environment of the automotive sector, product-oriented manufacturing systems are typical. In these environments, the manufacture of a product (engine, stamping part, framework, welded bodywork, painted bodywork, chassis, dress, etc.) is conceived as a set of consecutive stages or processes that add value by converting raw materials into finished products (vehicle). These manufacturing processes can be highly automated using robots or require a large number of human resources owing to the complexity of some operations.

Product-oriented manufacturing leads to flexible production systems composed of cells and the so-called production and assembly lines that are composed of modules or work stations.

Among the different types of production lines [Battaia and Dolgui (2013)], mixed-model assembly lines are found that can manufacture variants of a product (SUVs, 4 x 4, vans, etc.) with diverse options (long or short framework, sunroof or rigid roof, tinted or blind-metal window, etc.), without requiring substantial changes in staff or instruments in the work stations.

In the mixed-model production lines, two categories of problems can be distinguished:

- P1. Production line balancing: Problems oriented to the efficient allocation of a set of product assembly tasks to a set of workstations arranged in series, respecting a number of temporal (cycle time), spatial (available area) [Chica et al. (2016)] and contingency (ergonomic risk) constraints [Bautista et al. (2016)].
- P2. Sequencing of mixed products: Problems oriented to establishing the manufacturing order (entry to the line) of the product units according to one or more criteria, one or more demand plans, and a timeframe to execute them [Boysen et al. (2009)].

Owing to the complexity of both categories of problems, the standard practice, both in industry and academia, is to solve them consecutively: first, line balancing is established; subsequently, the most appropriate product sequence is determined in accordance with the imposed conditions.

The objectives, not necessarily excluded, that are considered when sequencing the models, respond to several productive concerns [Bautista and Cano (2011)]. Among the objectives are the following:

- o1 Maximize the number of units completed in the production line. Hence, efforts are conducted to reduce the inert time of the workers, the unnecessary waiting, and the production losses caused by workload excess (overload) in the stations [Yano and Rachamadugu (1991), Cano-Belmán et al. (2010), Bautista et al. (2012)].
- o2 Maximize the satisfaction of productive restrictions. These restrictions are related to the critical components of the vehicles or to the standard options included in the model catalogue [Parrello et al. (1986), Bautista et al. (2008.a)].



Joaquín Bautista-Valhondo  
[joaquin.bautista@upc.edu](mailto:joaquin.bautista@upc.edu)  
 ORCID: 0000-0002-2214-4991

Institut d'Organització i Control de Sistemes Industrials  
 (IOC - Divisió de Robòtica).  
 Universitat Politècnica de Catalunya, Av. Diagonal 647,  
 08028 Barcelona (Spain)

03 Minimize the stock levels of the components, both in the production plant and in the supply chain. Given the regularity in the production [Bautista and Alfaro (2017)], it is intended to maximize the consistency of the production rates of products [Bautista et al. (1997), Corominas and Moreno (2003)] and the rates of component consumption [Monden (1994), Bautista et al. (1996)].

Among the sequencing problems corresponding to objective (o2), the car sequencing problem (CSP) can be found, whose original purpose [Parrello et al. (1986)] consists in to establish a sequence  $\pi(T)$  of different type  $T$  vehicles in function of their optional elements. The nature of the problem is highly combinatorial [Gent (1998), Kis (2004)], for which its resolution has traditionally focused on the use of metaheuristics [Gottlieb et al. (2003), Bautista et al. (2008.b), Ribeiro et al. (2008), Morin et al. (2009), Siala et al. (2015)].

In the CSP, the vehicles are classified in a set  $I$ , of which  $d_i$  are of the type  $i$  ( $\forall i \in I$ ). Obviously, the sequence  $\pi(T)$  must be constructed in harmony with the vehicle demand plan that is symbolized using the vector  $d^T = (d_1, \dots, d_{|I|})$ , satisfying  $D \equiv T = \sum_{\forall i} d_i$ .

The typology of these vehicles is a function of the presence of optional elements - classified in a set  $J$ -. Therefore, a vehicle type  $i$  ( $i=1, \dots, |I|$ ) may or may not contain the option  $j$  ( $j=1, \dots, |J|$ ); this is reflected by the parameters  $n_{j,i}$  that adopt the value 1 if the option  $j \in J$  is present in the vehicle type  $i \in I$ , and the value 0 otherwise.

These optional elements are the focus of the problem, because their requirements by a group of vehicles added consecutively to the production line is limited. These limitations can be represented through the ratios  $p_j/q_j$  ( $\forall j \in J$ ) that symbolize the following: given the option  $j \in J$ , its requirement by the vehicles contained in any segment of the sequence  $\pi(T)$  with a length  $q_j$  (i.e., every segment with  $q_j$  consecutive production cycles) must be less than or equal to the value  $p_j$ ; colloquially, as a maximum, the option  $j \in J$  will be present  $p_j$  times in each cycle  $q_j$ .

Under these conditions, the original CSP consists of obtaining a sequence  $\pi(T)$  that satisfies all the restrictions on the optional element requirements. If the above is impossible, the objective of the problem is to ensure that the sequence  $\pi(T)$  satisfies the greatest number of such restrictions.

The remainder of this paper is structured as follows. In section 2, the new problem incorporating fleets of special vehicles and robustness to the original CSP are presented, where some peculiarities of these fleets are described. In section 3, the working hypotheses for the problem are stated, a nomen-

clature is presented, a basic optimization model based on the mixed integer linear programming (MILP) is proposed, and the multi-sequence concept resulting from the exploitation of the proposed model is introduced. In section 4, several metrics to evaluate the non-robustness and robustness of a multi-sequence of vehicles is offered. In section 5, the elementary variants on the r-CSP basic model (robust-CSP) are gathered: mono-objective functions based on the weighting of the excess of optional elements requirements and in the robustness metrics of a of production multi-sequence are proposed, as well as functions to address the multi-objective optimization in the problem. Finally, section 6 presents a synthesis and some considerations regarding this work.

## 2. CSP with fleets of special vehicles

Large automotive companies contain manufacturing and assembly lines that can accommodate their production programs, e.g., the insertion of non-regular or out-of-catalogue vehicles.

These special vehicles, termed fleets herein, are distinguished from the typical or regular vehicles for various reasons, and present some peculiarities:

- They are vehicles with demand under contract and are not part of the sales forecasts prepared by the commercial department.
- Their destination varies and typically corresponds to public service organizations: ambulances, vehicles destined to defense, armed forces vehicles, fire trucks, police patrol cars, forest guard vehicles, etc.
- These fleets require non-typical components that, when incorporated to the vehicles in the production line, generate additional operations that require greater processing times than standard operations.
- The non-typical components of special vehicles also depend on the type of fleet (ambulance, police patrol cars, forest guard vehicles, etc.). Therefore, the incorporation of a special vehicle type to the daily production program will cause differences in the consumption of components, for the requirement of tools and equipment, in the workloads and in the line supplies.
- Logically, the inclusion of special vehicles in the existing production programs will be limited by a number or by a proportion of those over the total. Typically, this proportion oscillates between 10% and 20% of the daily production of vehicles.

- f. To facilitate the production management, the daily global manufacture of special vehicles is set at a constant value; however, the partial production of each fleet type may vary by day. Meanwhile, the production of typical vehicles is not subject to daily demand alterations; in fact, their daily demand is stable and regular, and is aimed at satisfying the weekly or monthly demand plan.
- g. The possible variation in the daily demand of the special vehicles fleets generates uncertainty when preparing the production schedule; consequently, the manufacturing sequence of vehicles will be subjected to uncertainty. Obviously, it is impossible to establish a manufacturing sequence if the elements that constitute it are unknown.
- h. An alternative to handle the demand variation in the special vehicles fleets is to assume different demand scenarios. These scenarios must be realistic and in accordance with the production capacity and history of the manufacturing plant.

Formally, the demand scenarios will be grouped into a set  $E$  of elements  $\epsilon \in E$ . To define the scenario  $\epsilon \in E$ , the demand vector  $\overline{d_\epsilon} = (d_{1\epsilon}, \dots, d_{|I|\epsilon})$ , and the mixed-production vector  $\overline{\lambda_\epsilon} = (\lambda_{1\epsilon}, \dots, \lambda_{|I|\epsilon})$  will be used, where  $d_{i\epsilon}$  and  $\lambda_{i\epsilon}$  are the number of vehicles  $i \in I$  and their proportions in the plan  $\epsilon \in E$ , respectively. For coherence reasons,  $\overline{\lambda_\epsilon} = \overline{d_\epsilon} / D_\epsilon$  y  $D_\epsilon = \sum_{i \in I} d_{i\epsilon}$ , must be satisfied.

It is noteworthy that the ordinary CSP consists in obtaining a sequence  $\pi(T)$  that satisfies the largest number of restrictions on the optional elements requirements, while the CSP with fleets of special vehicles, which will be termed r-CSP herein, consists of obtaining a set of sequences  $\pi_\epsilon(T)$ , one for each plan  $\epsilon \in E$ , that satisfy the largest number of such restrictions; in addition, these sequences resemble each other closely [Bautista (2016)].

### 3. Basic CSP model with fleets of special vehicles

This section describes the proposal of a basic model for the CSP with fleets of special vehicles: a problem that will also be referred to as a robust version of the CSP (r-CSP).

First, the working hypotheses about the new problem is stated; next, a nomenclature to define and describe the variables and parameters that intervene in the model is proposed; subsequently, the r-CSP model is formulated within the framework of MILP; finally, the concept of multi-sequence production is defined as a solution representation form of any instance of the problem, for a set of demand plans that must be satisfied simultaneously as much as possible.

#### HYPOTHESIS:

1. Two families of vehicles are available: (i) the family of typical or regular vehicles, represented by the set of types  $I_x$ ; (ii) the family of special vehicles fleets, represented by the set of types  $I_x$ .
2. The total number of regular vehicles  $D_x$ , corresponding to a working day, is identical for all the demand plans  $\epsilon \in E$ .
3. The total number of special fleet vehicles  $D_x$ , corresponding to a working day, is identical for all the demand plans  $\epsilon \in E$ .
4. Consequently, the total number of vehicles  $T$  ( $T \equiv D = D_x + D_x$ ) that corresponds to a working day, is identical for all the demand plans  $\epsilon \in E$ .
5. The demand for a regular vehicle  $i \in I_x$ , corresponding to a working day, is identical for all the demand plans  $\epsilon \in E$ . That is, if  $i \in I_x$ , then  $d_{i\epsilon} = d_i \quad \forall \epsilon \in E$  is satisfied.
6. The demand for a special fleet vehicle  $i \in I_x$ , corresponding to a working day, may change for two different demand plans  $\{\epsilon, \epsilon'\} \subseteq E$ .
7. To minimize the number of changes in the production line (e.g., robots, instruments, tools, shelves, personnel, etc.), the manufacturing sequences  $\pi_\epsilon(T)$  and  $\pi_{\epsilon'}(T)$  must be as similar as possible for every pair of plans  $\{\epsilon, \epsilon'\} \subseteq E$ . Symbolically,  $\pi_\epsilon(T) \approx \pi_{\epsilon'}(T) \quad \forall \{\epsilon, \epsilon'\} \subseteq E$ .
8. As an immediate consequence of the previous hypothesis, all regular vehicles will be forced to occupy individually and by type; the same positions apply in all the sequences  $\pi_\epsilon(T) (\forall \epsilon \in E)$ .

**Table 1** Nomenclature - parameters

Nomenclature	Parameters
$I_X$	Set of regular or standard types of vehicles ( $i=1, \dots,  I_X $ ).
$I_{X'}$	Set of special types of vehicles (fleets) ( $i= I_X +1, \dots,  I_X + I_{X'} $ ).
$I$	Set of vehicle types: $I=I_X \cup I_{X'}$ ( $i=1, \dots,  I $ ).
$J$	Set of component parts or optional elements ( $J=j=1, \dots,  J $ ).
$E$	Set of scenarios or demand plans ( $E:\varepsilon=1, \dots,  E $ ).
$\vec{d}_\varepsilon, D$	Demand plan vector $\varepsilon \in E$ : $\vec{d}_\varepsilon = (d_{1,\varepsilon}, \dots, d_{ I ,\varepsilon})$ and total demand of vehicles in a workday: $D = T = \sum_{\varepsilon \in E} d_{i,\varepsilon}$ , identical in all plans $\varepsilon \in E$ .
$\vec{\lambda}_\varepsilon$	Mixed-production plan vector $\varepsilon \in E$ : $\vec{\lambda}_\varepsilon = (\lambda_{1,\varepsilon}, \dots, \lambda_{ I ,\varepsilon})$ : $\vec{\lambda}_\varepsilon = \vec{d}_\varepsilon / D$
$n_{j,i}$	Binary parameter that adopts the value 1 if the optional element $j \in J$ is present in the vehicle type $i \in I$ , and the value 0 otherwise.
$p_j/q_j$	CSP ratios that symbolize the requirement of the option $j \in J$ by the vehicles contained in any segment of the sequences $\pi_\varepsilon(T)$ ( $\forall \varepsilon \in E$ ) with a length $q_j$ ; must be less than or equal to the value $p_j$ .
$c_{j,t,\varepsilon}$	Cost or weight attributed to the segment of consecutive cycles $[t-q_j+1, t]$ of the sequence $\pi_\varepsilon(T)$ , linked to the demand plan $\varepsilon \in E$ , when the requirement of the option $j \in J$ is greater than $p_j$ in that segment. In this basic model, all the unit costs will be assumed: $c_{j,t,\varepsilon} = 1$ ( $\forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E$ ).

**Table 2** Nomenclature - variables

Nomenclature	Variables
$\pi_\varepsilon(T)$	Complete sequence of vehicles $\pi_\varepsilon(T) = (\pi_{1,\varepsilon}, \dots, \pi_{T,\varepsilon})$ of the plan $\varepsilon \in E$ . The partial sequences of $\pi_\varepsilon(T)$ will be represented as $\pi_\varepsilon(t) = (\pi_{1,\varepsilon}, \dots, \pi_{t,\varepsilon}) \subseteq \pi_\varepsilon(T)$ , $\forall t \in [1, T]$ . The symbols $\pi_\varepsilon(t)$ y $\pi_\varepsilon(T)$ are also used as parameters.
$x_{i,t}$	Binary variable that adopts the value 1 if a regular vehicle unit $i \in I_X$ is assigned to the position $t$ ( $t=1, \dots, T$ ) of the sequences $\pi_\varepsilon(T)$ of the plan $\varepsilon \in E$ , and adopt the value 0 otherwise.
$x_{i,t,\varepsilon}$	Binary variable that adopts the value 1 if a special vehicle unit $i \in I_{X'}$ is assigned to the position $t$ ( $t=1, \dots, T$ ) of the sequence $\pi_\varepsilon(T)$ of the plan $\varepsilon \in E$ , and it is 0 otherwise.
$X_{i,t}$	Number of regular vehicle unit type $i \in I_X$ contained in all partial sequences $\pi_\varepsilon(t) \subseteq \pi_\varepsilon(T)$ of all plans $\varepsilon \in E$ . Its calculation is performed as follows: $X_{i,t} = \sum_{\varepsilon \in E} x_{i,t,\varepsilon}$ $\forall i \in I_X, \forall t \in [1, T]$
$X'_{i,t,\varepsilon}$	Number of special vehicle unit type $i \in I_{X'}$ contained in the partial sequence $\pi_\varepsilon(t) \subseteq \pi_\varepsilon(T)$ of the plan $\varepsilon \in E$ . Its calculation is performed as follows: $X'_{i,t,\varepsilon} = \sum_{t=1}^t x'_{i,t,\varepsilon}$ $\forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E$
$Y_{j,t,\varepsilon}$	Number of times that the option $j \in J$ is required by the regular and special vehicles contained in the partial sequence $\pi_\varepsilon(t) \subseteq \pi_\varepsilon(T)$ of the plan $\varepsilon \in E$ . It is calculated as follows: $Y_{j,t,\varepsilon} = \sum_{i \in I_X} n_{j,i} X_{i,t} + \sum_{i \in I_{X'}} n_{j,i} X'_{i,t,\varepsilon}$ $\forall j \in J, \forall t \in [1, T], \forall \varepsilon \in E$ . For convenience, it will be calculated as follows: $Y_{j,0,\varepsilon} = 0$ $\forall j \in J, \forall \varepsilon \in E$
$Z_{j,t,\varepsilon}$	Binary variable that adopts the value 1 if the requirement of option $j$ is greater than the value $p_j$ in the segment $[t-q_j+1, t]$ of the sequence $\pi_\varepsilon(T)$ , and it is 0 otherwise ( $\forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E$ ). For convenience, the symbols $Z_{j,t,\varepsilon}$ will also be used as parameters when the sequence $\pi_\varepsilon(T)$ is known.



FORMULATION - BASIC r-CSP MODEL:

$$\min Z = \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} z_{j,t,\varepsilon} \Leftrightarrow \max Z' = \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} (1 - z_{j,t,\varepsilon}) \quad (1)$$

Subject to:

$$\sum_{i \in I_X} x_{i,t} + \sum_{i \in I_{X'}} x'_{i,t,\varepsilon} = 1 \quad \forall t \in [1, T], \forall \varepsilon \in E \quad (2)$$

$$\sum_{t \in [1, T]} x_{i,t} = d_i \quad \forall i \in I_X \quad (3)$$

$$\sum_{t \in [1, T]} x'_{i,t,\varepsilon} = d_{i,\varepsilon} \quad \forall i \in I_{X'}, \forall \varepsilon \in E \quad (4)$$

$$X_{i,t} - \sum_{\tau \in [1, t]} x_{i,\tau} = 0 \quad \forall i \in I_X, \forall t \in [1, T] \quad (5)$$

$$X'_{i,t,\varepsilon} - \sum_{\tau \in [1, t]} x'_{i,\tau,\varepsilon} = 0 \quad \forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E \quad (6)$$

$$Y_{j,t,\varepsilon} - \sum_{i \in I_X} n_{j,i} X_{i,t} - \sum_{i \in I_{X'}} n_{j,i} X'_{i,t,\varepsilon} = 0 \quad \forall j \in J, \forall t \in [1, T], \forall \varepsilon \in E \quad (7)$$

$$Y_{j,t,\varepsilon} - Y_{j,t-q_j,\varepsilon} \leq p_j + T \cdot z_{j,t,\varepsilon} \quad \forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E \quad (8)$$

$$x_{i,t} \in \{0,1\} \quad \forall i \in I_X, \forall t \in [1, T] \quad (9)$$

$$x'_{i,t,\varepsilon} \in \{0,1\} \quad \forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E \quad (10)$$

$$z_{j,t,\varepsilon} \in \{0,1\} \quad \forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E \quad (11)$$

$$Y_{j,0,\varepsilon} = 0 \quad \forall j \in J, \forall \varepsilon \in E \quad (12)$$

In the basic r-CSP model, the objective function (1) represents the minimization of the number of constraint violations that limit the requirements of any option  $j \in J$  to the maximum values  $p_j \forall j \in J$ , for each demand plan  $\varepsilon \in E$ , and for each interval of consecutive productive cycles of length  $q_j (\forall j \in J)$ . Equalities (2) impose the insertion of a vehicle to the line (regular or special), only one in each manufacturing cycle  $t \in [1, T]$ , and in every demand plan  $\varepsilon \in E$ . Equalities (3) and (4) force all the demand plans for the regular vehicles  $I_x$  and the special ones  $I_0$  to be satisfied, respectively. Equalities (5) are used to count the number of regular vehicles  $i \in I_x$  added to line until the manufacturing cycle  $t \in [1, T]$ , in any demand plan  $\varepsilon \in E$ . Meanwhile, equalities (6), analogous to (5), refer to the special vehicles  $I_x$ , considering each plan  $\varepsilon \in E$ . Equalities (7) count the number of times that option  $j \in J$  is required by the vehicles added consecutively to the line until a particular manufacturing cycle  $t \in [1, T]$  in any demand plan  $\varepsilon \in E$ . Restrictions (8) determine whether a violation of the requirements restriction of every option  $j \in J$  occurs in every segment, with a length  $q_j (\forall j \in J)$  of the sequence  $\pi_{\varepsilon}(T) (\forall \varepsilon \in E)$ . Conditions (9), (10), and (11) define  $x_{i,t}$ ,  $x'_{i,t,\varepsilon}$ , and  $z_{j,t,\varepsilon}$ , respectively, as binary variables. Finally, equalities (12) establish, for convenience, the variables of requirement  $Y_{j,0,\varepsilon} (\forall j \in J, \forall \varepsilon \in E)$  as null, in the fictitious production cycle  $t = 0$ .

The exploitation of the r-CSP model allows for a multi-sequence  $\vec{\pi}(E, T)$  to be obtained, and is composed of the sequences  $\pi_{\varepsilon}(T) = (\pi_{1,\varepsilon}, \dots, \pi_{T,\varepsilon})$  of each plan of demand  $\varepsilon \in E$ :

$$\vec{\pi}(E, T) = \left\{ \begin{matrix} \pi_1(T) \\ \pi_2(T) \\ \vdots \\ \pi_{\varepsilon}(T) \\ \vdots \\ \pi_{|E|}(T) \end{matrix} \right\} = \left\{ \begin{matrix} (\pi_{1,1}, \dots, \pi_{t,1}, \dots, \pi_{T,1}) \\ (\pi_{1,2}, \dots, \pi_{t,2}, \dots, \pi_{T,2}) \\ \vdots \\ (\pi_{1,\varepsilon}, \dots, \pi_{t,\varepsilon}, \dots, \pi_{T,\varepsilon}) \\ \vdots \\ (\pi_{1,|E|}, \dots, \pi_{t,|E|}, \dots, \pi_{T,|E|}) \end{matrix} \right\} \quad [13]$$

The relationship between the vehicle type  $i \in I$  and the elements  $\pi_{i,\varepsilon} (\forall t \in [1, T], \forall \varepsilon \in E)$  of the multi-sequence  $\vec{\pi}(E, T)$ , is established through the values adopted by the binary variables  $x_{i,t} (\forall i \in I_x, \forall t \in [1, T])$  y  $x'_{i,t,\varepsilon} (\forall i \in I_x', \forall t \in [1, T], \forall \varepsilon \in E)$ , present in the r-CSP model. Therefore,

$$x_{i,t} = 1 \Rightarrow \pi_{t,\varepsilon} = i, \forall i \in I_x, \forall t \in [1, T], \forall \varepsilon \in E \quad [14]$$

$$x'_{i,t,\varepsilon} = 1 \Rightarrow \pi_{t,\varepsilon} = i, \forall i \in I_x', \forall t \in [1, T], \forall \varepsilon \in E \quad [15]$$

It is noteworthy that the regular vehicles ( $i \in I_x$ ) occupy all the same positions, by type and by cycle, in all the sequences of all the demand plans, while the positions occupied by the fleets depend on each plan  $\varepsilon \in E$ . In other words, all the sequences  $\pi_{\varepsilon}(T)$  will present a common part (composed by regular vehicles) and an exclusive part (composed by special vehicles).

## 4. Metrics for the robustness of a multi-sequence r-CSP

The r-CSP formulation described in the previous section corresponds to a problem of maximum satisfaction of restrictions (MAXSAT), such that its connection with the genuine CSP of Parrello, Kabat, and Vos is clear. Further, it is clear that techniques based on automatic reasoning can be used for its resolution.

Far from being satisfied with presenting and solving a MAXSAT problem and assisted by the multi-sequence production concept  $\vec{\pi}(E, T)$ , in this section, several methods of measuring the quality of any solution are proposed, based on the following:

Definition 1: The multi-sequence  $\vec{\pi}(E, T)$  is strongly robust against the triple  $(I, J, p_j / (q_j, E))$ , when  $z_{j,t,\varepsilon} = 0 \forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E$  is satisfied.

When  $\vec{\pi}(E, T)$  does not satisfy the restrictions  $p_j / q_j (\forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E)$ , then its quality will be evaluated through the following non-robustness metrics:

- m1. Proportion of demand plans that present excessive requirement of optional elements ( $j \in J$ ) in a particular manufacturing cycle ( $t \in [q_j, T]$ ) by the multi-sequence  $\vec{\pi}(E, T)$ . It is useful for detecting critical demand plans.

$$g_1(\vec{\pi}(E, T)) = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \max_{\forall j \forall t} \{z_{j,t,\varepsilon}\} \quad [16]$$

- m2. Proportion of options of set  $J$  that are required in excess by the multi-sequence  $\vec{\pi}(E, T)$  in a particular manufacturing cycle ( $t \in [q_j, T]$ ) and in a particular demand plan of the set  $E$ . It is useful for detecting critical optional elements.

$$g_2(\vec{\pi}(E, T)) = \frac{1}{|J|} \sum_{j=1}^{|J|} \max_{\forall t \forall \varepsilon} \{z_{j,t,\varepsilon}\} \quad [17]$$

- m3. Proportion of manufacturing cycles with excessive requirement of optional elements ( $j \in J$ ) by the multi-sequence  $\vec{\pi}(E, T)$  in a particular demand plan of the set  $E$ . It is useful to detect critical production cycles.

$$g_3(\vec{\pi}(E, T)) = \frac{1}{T} \sum_{t=1}^T \max_{\forall j \forall \varepsilon} \{z_{j,t,\varepsilon}\} \quad [18]$$

m4. Proportion of constraints  $p_j/q_j$  ( $\forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E \forall j \in J$ ) that violates the multi-sequence  $\vec{\pi}(E, T)$ . It also represents the average proportion of manufacturing cycles with the excess of optional elements requirements ( $j \in J$ ) in the set of demand plans  $E$ . It is useful to determine the global non-robustness of the sequence  $\vec{\pi}(E, T)$  against all the constraints of the problem. A lower bound of metric m.4 is as follows:

$$\hat{g}_4(\vec{\pi}(E, T)) = \frac{1}{|E| \cdot |J| \cdot T} \sum_{j=1}^{|J|} \sum_{t=1}^T \sum_{\varepsilon=1}^{|E|} z_{j,t,\varepsilon} \quad [19]$$

The refined metric function  $g_4(\vec{\pi}(E, T))$  that strictly counts the restrictions that act on the productive cycles  $t \in [q_j, T]$  ( $\forall j \in J$ ), is defined as follows:

$$g_4(\vec{\pi}(E, T)) = \frac{1}{|E| \sum_{j=1}^{|J|} (T + 1 - q_j)} \sum_{j=1}^{|J|} \sum_{t=1}^T \sum_{\varepsilon=1}^{|E|} z_{j,t,\varepsilon} \quad [20]$$

m5. Proportion of manufacturing cycles with maximum excess of optional elements requirements ( $j \in J$ ) among the set of demand plans  $E$ . It is useful to determine the most critical demand plan.

$$g_5(\vec{\pi}(E, T)) = \max_{\forall \varepsilon} \left\{ \frac{1}{\sum_{j=1}^{|J|} (T + 1 - q_j)} \sum_{j=1}^{|J|} \sum_{t=1}^T z_{j,t,\varepsilon} \right\} \quad [21]$$

m6. Proportion of manufacturing cycles with the maximum excess requirements among the optional elements ( $j \in J$ ) in the set of demand plans  $E$ . It is useful to detect the most critical optional element.

$$g_6(\vec{\pi}(E, T)) = \max_{\forall j} \left\{ \frac{1}{T \cdot |E|} \sum_{t=1}^T \sum_{\varepsilon=1}^{|E|} z_{j,t,\varepsilon} \right\} \quad [22]$$

m7. Proportion of optional elements ( $j \in J$ ) with excessive requirements in all the demand plans of the set  $E$ , which corresponds to the last production cycle of the most critical segments of the multi-sequence  $\vec{\pi}(E, T)$ . It is useful to detect, on average, the most critical manufacturing cycle.

$$g_7(\vec{\pi}(E, T)) = \max_{\forall t} \left\{ \frac{1}{|J| \cdot |E|} \sum_{j=1}^{|J|} \sum_{\varepsilon=1}^{|E|} z_{j,t,\varepsilon} \right\} \quad [23]$$

From the previous non-robustness metrics, it is possible to instantly define their corresponding metrics to measure the robustness of  $\vec{\pi}(E, T)$  against the triple  $(I, J, p_j(q_j), E)$ . That is,

$$r_m(\vec{\pi}(E, T)) = 1 - g_m(\vec{\pi}(E, T)) \quad m = 1, \dots, 7 \quad [24]$$

It is noteworthy that thus far, the terms "excessive requirements" or "excess of requirements" have been used interchangeably to reflect the violation of a particular restriction  $p_j/q_j$  (see restrictions (8) of the r-CSP model), without indicating how its amount or cost is to be determined. These possible extensions of the basic r-CSP will be exposed in the

subsequent work.

## 5. Elementary variants of the basic r-CSP model

Logically, the basic r-CSP model supports several variants, either by simplification or by its elementary extension. Some of them are presented below:

- Simplification: Obviously, if only one demand plan exists (i.e.,  $|E|=1$ ), the basic r-CSP model becomes the original CSP model. Consequently, it can be understood that the CSP is a particular case of the r-CSP; therefore, the optimal solutions of the CSP will be useful to calculate the lower bounds of the r-CSP.
- Weighting of the excess requirement of options: If costs or weights for excess requirements ( $c_{j,t,\varepsilon}$ ) are considered, the objective function (1) of the basic r-CSP model must be replaced by the following:

$$\min \hat{Z} = \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} c_{j,t,\varepsilon} \cdot z_{j,t,\varepsilon} \quad [25]$$

where  $\hat{Z}$  symbolizes the total cost for the requirement excess of optional elements of set  $J$  in the set of demand plans  $E$ . Here, such total cost is evaluated as a weighted sum of violations of the restrictions  $p_j/q_j$  of the r-CSP.

- Mono-objective models with robustness functions of the multi-sequence: In case of incorporating the non-robustness metrics (m.1 to m.7) as elements of an optimization problem, the objective function (1) of the basic r-CSP must be replaced by one of the following functions:

$$\min Z = g_m(\vec{\pi}(E, T)) \Leftrightarrow \max Z' = r_m(\vec{\pi}(E, T)) \quad m = 1, \dots, 7 \quad [26]$$

- Bi-objective models with multi-sequence robustness functions: Obviously it is necessary to formulate the bi-objective optimization models of robustness, replacing the objective function (1) of the basic r-CSP with any of the following functions:

$$\left\{ \begin{array}{l} \min g_1(\vec{\pi}(E, T)) \wedge \min g_2(\vec{\pi}(E, T)) \\ \min g_1(\vec{\pi}(E, T)) \wedge \min g_3(\vec{\pi}(E, T)) \\ \min g_2(\vec{\pi}(E, T)) \wedge \min g_3(\vec{\pi}(E, T)) \end{array} \right\} \quad [27]$$



For example, plan # 8 is represented by the following demand plan vector:

$$\vec{d}_8=(5,3,7,1,10,2,11,5,4,6,12,1,1,5,9,5,12,1)$$

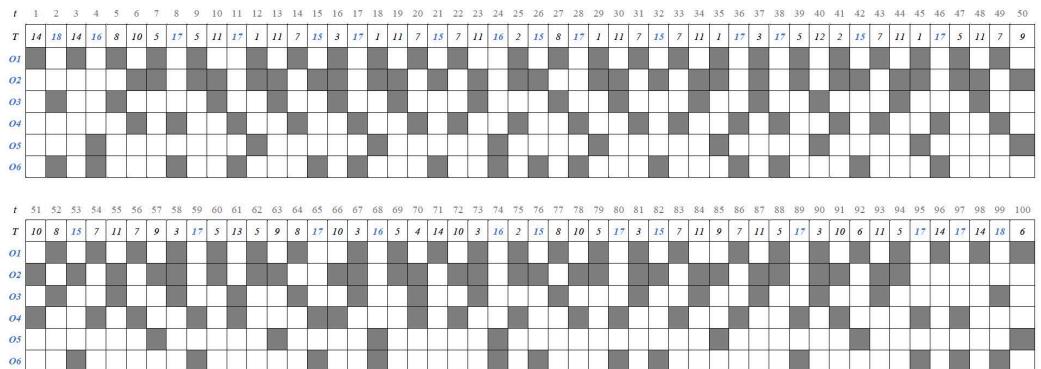
This plan is identical to that of the original instance with a total demand  $D=100$ . Figure 2 shows a solution to the problem that satisfies all restrictions.

Similarly, Figures 3 and 4 show, respectively, two possible sequences for instances # 1 and # 22, whose demand plan vectors are:

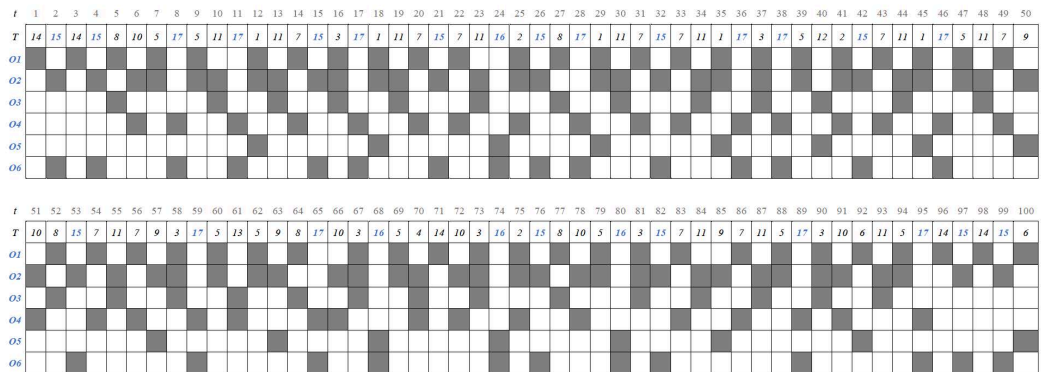
$$\vec{d}_1=(5,3,7,1,10,2,11,5,4,6,12,1,1,5,8,4,13,2)$$

$$\vec{d}_{22}=(5,3,7,1,10,2,11,5,4,6,12,1,1,5,12,4,11,0)$$

**Figure 3** Solution of the instance #1, considering  $p_j/q_j = 1/2$  (for  $j=6$ ) and  $d_{15,1}=8, d_{16,1}=4, d_{17,1}=13$  and  $d_{18,1}=2$ . The total demand of vehicles in a workday is  $D=100$  ( $D_x=73, D_y=27$ ).



**Figure 4** Solution of the instance #22, considering  $p_j/q_j = 1/2$  (for  $j=6$ ) and  $d_{15,22}=12, d_{16,22}=4, d_{17,22}=11$  and  $d_{18,22}=0$ . The total demand of vehicles in a workday is  $D=100$  ( $D_x=73, D_y=27$ ).



Given the conditions of the r-CSP, the demands of the regular vehicles (1 to 14) of the three previous demand plans (# 1, # 8 and # 22) are identical one by one, while the demands of the vehicles of fleet (15 to 18) may vary from one plan to another.

In any case, the global demand for fleet vehicles is always the same ( $D_x$ )=27 cars). This condition is imposed on the 22 demand plans included in Table 1 of Annex-I.

The above observation is the main characteristic of the robust car sequencing problem: the demands of the regular vehicles are fixed in all plans, while the individual demands of fleet vehicles can vary.

In summary, our example consists of finding a sequence of regular vehicles of type 1 to 14 in fixed positions (see Figure 5), to which is added a multi-sequence of fleet vehicles of type 15 to 18, so that all CSP-ratios  $p_j/q_j$  ( $j \in J$ ) are satisfied.

**Figure 5** Map of options for the sequences of regular vehicles of type 1 to 14 in the 22 demand plans. Free columns (positions) for fleet vehicles are headed by the letter "F".

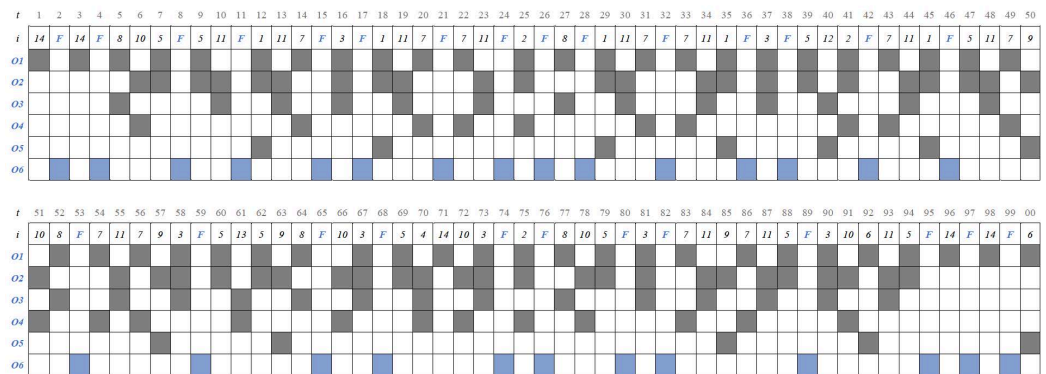


Table 2 of Annex-I shows the multi-sequence of the fleet vehicles (15 to 18) for the 22 production plans considered. All sequences satisfy the maximum load restrictions for options O1 to O6.

### 7. Synthesis and final considerations

This paper presents a new problem under the term, “car sequencing problem” with fleets of special vehicles and the acronym r-CSP (robust-CSP). After introducing the concept of partial demand uncertainty in special vehicle fleets with its peculiarities and present-ing the hypothesis of the problem, an optimization model based on MILP was formulat-ed, whose exploitation results in a multi-sequence of manufacturing.

The multi-sequence definition allowed for the concept of robustness to be incorporated in the sequencing problems of mixed models with partial demand uncertainty. Regard-ing the specific case of the r-CSP, 7 metrics to evaluate the non-robustness (robustness) of a solution have been proposed and could also be used as objective functions yielding several mono and multi-objective variants of the optimization problem.

The dimensions of the proposed optimization models are on the order of 23000 binary variables and 38000 explicit restrictions, considering industrial instances with 20 types of regular vehicles, 5 types of fleet vehicle, 10 types of optional elements, and 10 demand plans with 135 vehicles in a work-day. Although these dimensions are suitable to obtain solu-tions through MILP, it is convenient and advisable to also use metaheuristics to solve the r-CSP.

Furthermore, the proposals included herein could be incorporated into other sequencing problems of mixed-model production lines, or into other scheduling problems, when the appropriate circumstances arise.

**Acknowledgments.** This work has been funded by the Ministry of Economy and Competitiveness (Government of Spain) though the FHI-SELM2 (ref. TIN2014-57497-P) and OPTHEUS (ref. PGC2018-095080-B-I00), projects, including European Regional Development Funds (ERDF).

### 8. References

Battaia, O., Dolgui, A. (2013). “A taxonomy of line balancing problems and their solution approaches”. *International Journal of Production Economics* 142(2), pp 259-277. DOI: 10.1016/j.ijpe.2012.10.020

Bautista, J. (2016). “Modelos y métricas para la versión robusta del Car Sequencing Problem con Flotas de vehículos especiales” *Dirección y Organización* 60 (Diciembre 2016), pp 57-65. <https://www.revistadyo.es/index.php/dyo/article/view/499>

Bautista, J., Alfaro, R. (2017). “Free and regular mixed-model sequences by a linear program-assisted hybrid algorithm GRASP-LP”. *Progress in Artificial Intelligence* 6(2), pp 159-169. DOI: 10.1007/s13748-017-0110-z

Bautista, J., Batalla-García, C., Alfaro-Pozo, R. (2016). “Models for assembly line balancing by temporal, spatial and ergonomic risk attributes”. *European Journal of Operational Research* 251(3), pp 814-829. DOI: 10.1016/j.ejor.2015.12.042

Bautista, J., Cano, A. (2011). “Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimization and interruption rules”. *European Journal of Operational Research* 210(3), pp 495–513. DOI: 10.1016/j.ejor.2010.10.022

Bautista, J., Cano, A., Alfaro, A. (2012). “Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints. An application in Nissan’s Barcelona Plant”. *Expert Systems with Applications* 39 (2012), pp 11001–11010. DOI: 10.1016/j.eswa.2012.03.024

Bautista, J., Companys, R., Corominas, A. (1996). “Heuristics and exact algorithms for solving the Monden problem”. *European Journal of Operational Research* 88(1), pp 101-113. DOI: 10.1016/0377-2217(94)00165-0

Bautista, J., Companys, R., Corominas, A. (1997). “Modelling and solving the production rate variation problem (PRVP)”. *Top* 5(2), pp 221–239. DOI: 10.1007/BF02568551

- Bautista, J., Pereira, J., Adenso-Díaz, B. (2008.a). "A GRASP approach for the extended car sequencing problem". *Journal of Scheduling* 11, pp 3-16. DOI: 10.1007/s10951-007-0046-4
- Bautista, J., Pereira, J., Adenso-Díaz, B. (2008.b). "A beam search approach for the optimization version of the car sequencing problem". *Annals of Operations Research* 159(1), pp 233–244. DOI: 10.1007/s10479-007-0278-x
- Boysen, N., Fliedner, M., Scholl, A. (2009). "Sequencing mixed-model assembly lines: Survey, classification and model critique". *European Journal of Operational Research* 192(2), pp 349–373. DOI: 10.1016/j.ejor.2007.09.013
- Cano-Belmán, J., Ríos-Mercado, R.Z., Bautista, J. (2010). "A scatter search based hyper-heuristic for sequencing a mixed-model assembly line". *Journal of Heuristics* 16(6), pp 749-770. DOI: 10.1007/s10732-009-9118-2
- Chica, M., Bautista, J., Cordon, O., Damas, S. (2016). "A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand". *Omega* 58, pp 55-68. DOI: 10.1016/j.omega.2015.04.003
- Corominas, A., Moreno, N. (2003). "On The Relations Between Optimal Solutions For Different Types Of Min-Sum Balanced Jit Optimisation Problems". *INFOR: Information Systems and Operational Research* 41(4), pp 333-339. DOI: 10.1080/03155986.2003.11732685
- Gent, I.P. (1998). "Two results on car-sequencing problems". Report University of Strathclyde, APES-02-98.
- Gottlieb, J., Puchta, M., Solnon, C. (2003). "A study of greedy, local search, and ant colony optimization approaches for car sequencing problems". In Stefano Cagnoni, S. et al. (eds.): *Applications of evolutionary computing*, pp 246-257. Ed. Springer. DOI: 10.1007/3-540-36605-9\_23
- Kis, T. (2004). "On the complexity of the car sequencing problem". *Operations Research Letters* 32 (2004), pp 331-335. DOI: 10.1016/j.orl.2003.09.003
- Little, J. (1993). "A searching technique!". *OR Insight*, 6(4), pp 24–31. DOI: 10.1057/ori.1993.25
- Monden, Y. (1994). *Toyota Production System. An Integrated Approach to Just-In-Time*. Ed. Springer US. DOI: 10.1007/978-1-4615-9714-8
- Morin, S., Gagné, C., Gravel, M. (2009). "Ant colony optimization with a specialized pheromone trail for the car-sequencing problem". *European Journal of Operational Research* 197(3), pp 1185-1191. DOI: 10.1016/j.ejor.2008.03.033
- Parrello, B.D., Kabat, W.C., Wos, L. (1986). "Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem". *Journal of Automated reasoning* 2(1), pp 1-42. DOI: 10.1007/BF00246021
- Ribeiro, C.C., Aloise, D., Noronha, T.F., Rocha, C., Urrutia, S. (2008). "An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem". *European Journal of Operational Research* 191(3), pp 596-611. DOI: 10.1016/j.ejor.2007.02.003
- Siala, M., Hebrard, E., Huguet, M.J. (2015). "A study of constraint programming heuristics for the car-sequencing problem". *Engineering Applications of Artificial Intelligence* 38 (2015), pp 34–44. DOI: 10.1016/j.engappai.2014.10.009
- Yano, C.A., Rachamadugu, R. (1991). "Sequencing to Minimize Work Overload in Assembly Lines with Product Options". *Management Science* 37(5), pp 572-586. DOI: 10.1287/mnsc.37.5.572 Annex-I

**Annex-I**

**Table 1** Demand plans ( $\epsilon \in E$ ) for fleet vehicles of type 15 to 18 with a joint demand of 27 cars. The demand for regular vehicles is identical in all plans: 5, 3, 7, 1, 10, 2, 11, 5, 4, 6, 12, 1, 1, 5 ( $D_x=73$ ).

$i \setminus \epsilon$	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22
15	8	8	9	9	9	9	9	9	10	10	10	10	10	10	10	11	11	11	11	11	12	12
16	4	4	3	4	4	4	5	5	3	3	4	4	4	5	5	3	3	4	4	5	3	4
17	13	14	14	12	13	14	11	12	13	14	11	12	13	11	12	12	13	11	12	11	12	11
18	2	1	1	2	1	0	2	1	1	0	2	1	0	1	0	1	0	1	0	0	0	0
Total	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27

**Table 2** Multi-sequence of fleet vehicles of type 15 to 18 corresponding to the 22 demand plans ( $\epsilon \in E$ ). The columns of the table are headed by the positions (t) that fleet vehicles occupy in the 22 sequences.

$\epsilon \setminus t$	2	4	8	11	15	17	21	24	26	28	32	36	38	42	46	53	59	65	68	74	76	80	82	89	95	97	99
#1	18	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	18
#2	17	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	18
#3	17	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	18
#4	18	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	18
#5	18	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	15
#6	17	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	15
#7	18	16	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	18
#8	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	17	18
#9	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	18
#10	17	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	17	15
#11	18	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	18
#12	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	17	18
#13	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	17
#14	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	18
#15	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	17
#16	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	18
#17	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	17
#18	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	18
#19	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	15
#20	16	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	15
#21	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	17	15	17	17	15	15
#22	15	15	17	17	15	17	15	16	15	17	15	17	17	15	17	15	17	17	16	16	15	16	15	17	17	15	15