Networking and Emerging Optimization

NEO Research Group

http://neo.lcc.uma.es

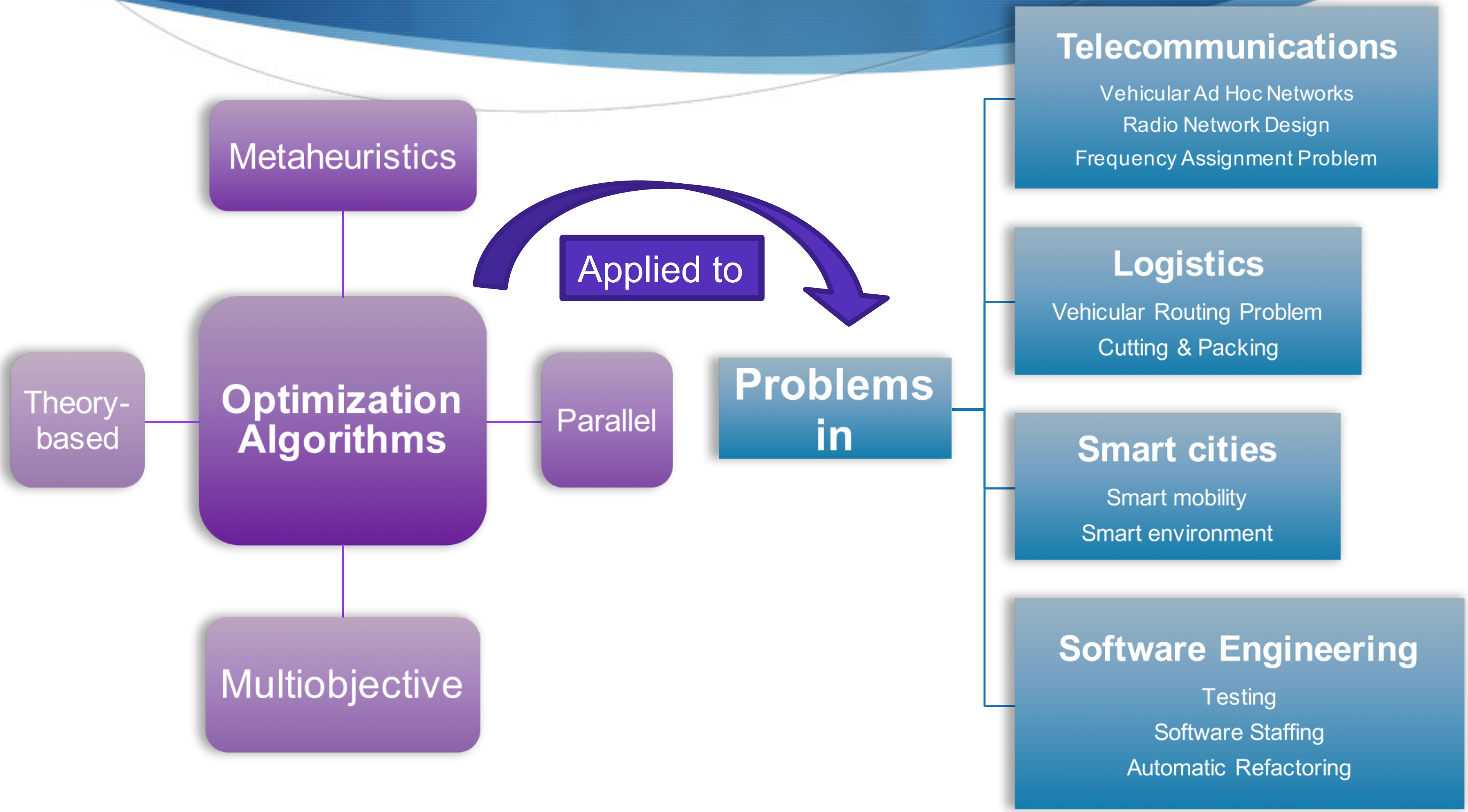
GISUM

UNIVERSIDAD DE MÁLAGA

Members

NEO Research Group

http://neo.lcc.uma.es



Our Research in SBSE

- Software Project Scheduling
- Requirements Selection
- Automatic Refactoring
- White-box Software Testing
- Testing of Concurrent Systems (based on Model Checking)
- Testing Complexity
- Prioritized Pairwise Combinatorial Interation Testing
- Test Sequences for Functional Testing
- Test Suite Minimization
- Software Product Lines Testing

JISBD
XX Jornadas de Ingeniería del Software y Bases de Datos

Resolviendo un problema multi-objetivo de selección de requisitos mediante resolutores del problema SAT

UNIVERSIDAD DE ALMERÍA
UNIVERSIDAD DE MÁLAGA

Isabel del Águila, José del Sagrado, Francisco Chicano y Enrique Alba

Problemas NP-difíciles

En muchos artículos se puede leer…

"Este problema de optimización es NP-difícil y, por este motivo, usamos…

- **Metaheurísticas**
- **Heurísticas**
- **Algoritmos estocásticos**

… que no aseguran una solución óptima, pero es capaz de encontrar buenas soluciones en un tiempo razonable

Hasta donde sabemos no existen algoritmos eficientes (tiempo polinomial) para resolver problemas NP-difíciles

Pero conocemos algoritmos ineficientes en el peor caso (tiempo al menos exponencial)

Problema SAT (de satisfacibilidad)

¿Podemos encontrar una asignación de valores lógicos (verdadero y falso) a variables tal que satisfaga un conjunto de fórmulas lógicas?

$$\neg A \wedge (B \vee C)$$
$$(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$$
$$A \vee B$$

Fue el primer problema NP-completo (Stephen Cook, 1971)

Si se pudiera resolver en tiempo polinomial P=NP

Los algoritmos conocidos lo resuelven en tiempo exponencial (en el peor caso)

Algoritmos del estado del arte en SAT

Actualmente, los resolutores SAT resuelven instancias con millones de variables

Esto es un espacio de búsqueda de $2^{1000\,000} \approx 10^{301030}$

Problema SAT (de satisfacibilidad)

¿Podemos usar los avances de los resolutores SAT para resolver problemas de optimización?

Mi problema favorito → Instancia de SAT → Resolutor SAT → Solución óptima

Selección de requisitos

Transformación a SAT

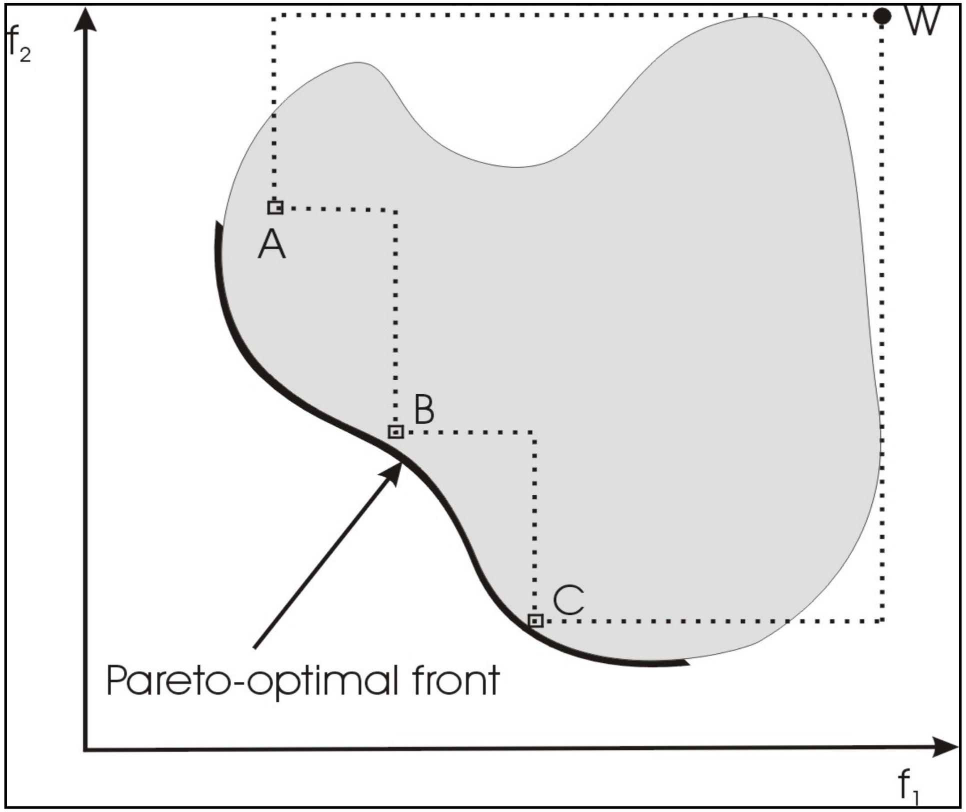
MiniSAT+

Algoritmo

Resultados

Problemas de Optimización Multi-Objetivo

En un problema MO hay varios objetivos (funciones) que queremos optimizar



Next Release Problem (NRP)

Dados:

- **Un conjunto de requisitos $R = \{r_1, r_2, \ldots, r_n\}$ …**
- **… cada uno con un coste c_j y un valor w_j**
- **Un conjunto de interacciones funcionales entre requisitos**
  - **Implicación (r_i antes que r_j):** $r_i \Rightarrow r_j$
  - **Combinación (r_i a la vez que r_j):** $r_i \odot r_j$
  - **Exclusión (no a la vez):** $r_i \oplus r_j$

Encontrar un subconjunto de requisitos $X \subseteq R$ que además de cumplir con las interacciones minimice el coste y maximice el valor:

$$\min \quad coste(X) = \sum_{j, r_j \in X}^{n} c_j$$

$$\max \quad valor(X) = \sum_{j, r_j \in X}^{n} s_j$$

Restricciones Pseudo-Booleanas

Una restricción Pseudo-Booleana (PB) tiene la forma:

$$\sum_{i=1}^{n} a_i x_i \odot B$$

donde

$$\odot \in \{<, \leq, =, \neq, >, \geq\}$$

$$a_i, B \in \mathbb{Z} \quad x_i \in \{0, 1\}$$

Se pueden transformar en instancias de SAT (normalmente eficiente)

Son un formalismo de alto nivel para especificar un problema de decisión

Pueden ser la entrada para MiniSAT+

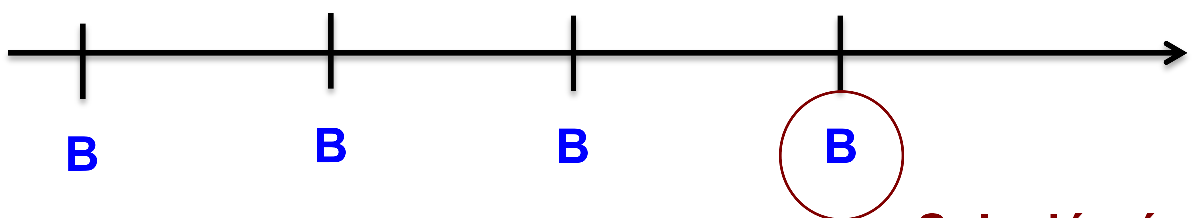
Transformación de Optimización a Decisión

Supongamos que queremos minimizar f(x)

Check **Check** **Check** **Check**

$$f(x) \leq B \quad f(x) \leq B \quad f(x) \leq B \quad f(x) \leq B$$

no **no** **no** **sí**



Solución óptima encontrada

Se puede hacer lo mismo en el caso de problemas multi-objetivo, pero necesitamos más restricciones PB (método *ε-contraint)*

$$f_1(y) \leq B_1 \qquad f_2(y) \leq B_2 \qquad \cdots \qquad f_m(y) \leq B_m$$

Transformación de NRP

- Implicación $r_i \Rightarrow r_j$: $r_j \leq r_i$.
- Combinación $r_i \odot r_j$: $r_i = r_j$.
- Exclusión $r_i \oplus r_j$: $r_i + r_j \leq 1$.

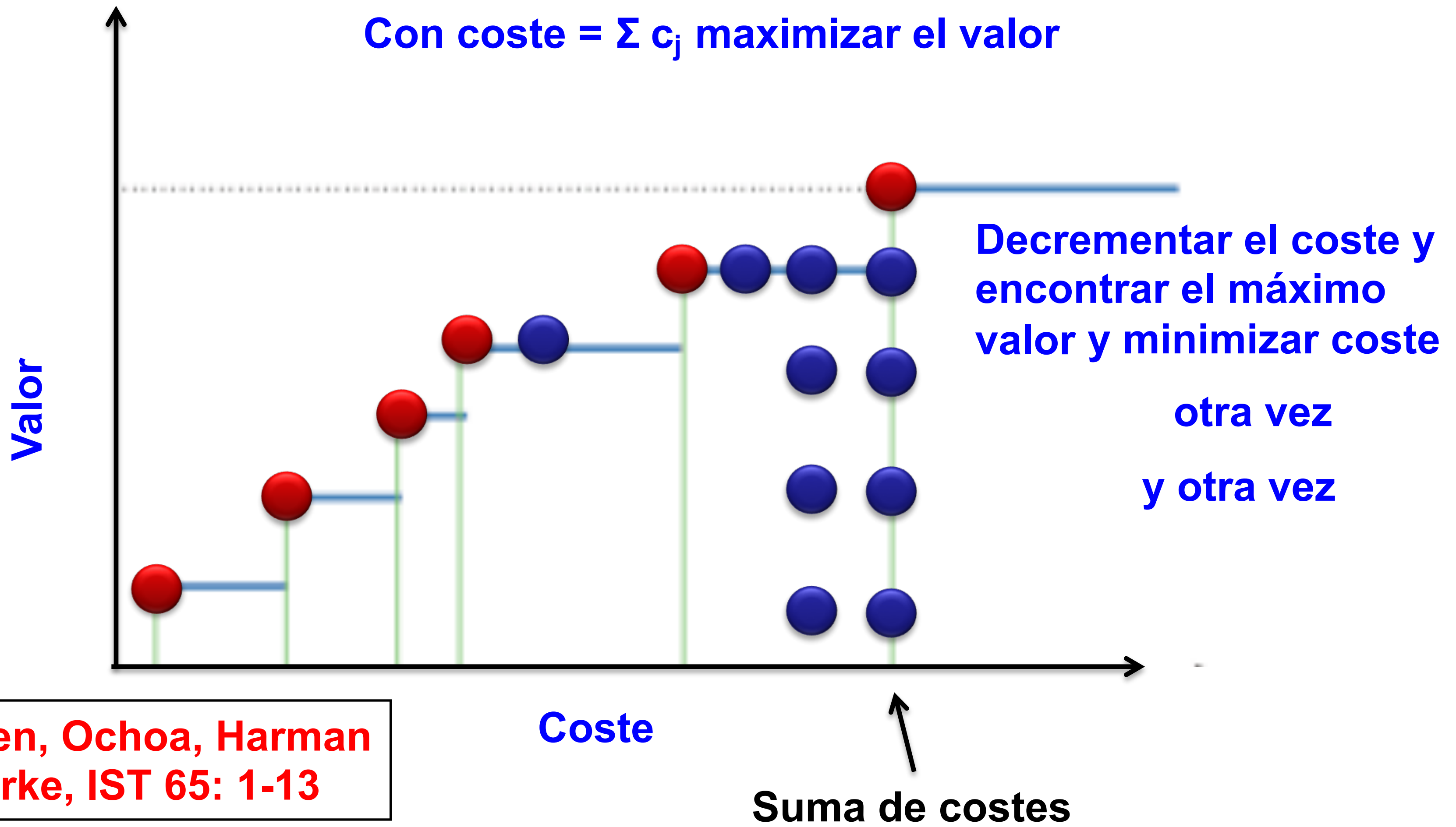
Minimizar coste

$$\text{mín} : \sum_{i=1}^{n} c_j r_j \quad \text{sujeto a:} \quad \sum_{j=1}^{n} s_j r_j \geq A.$$

Maximizar valor

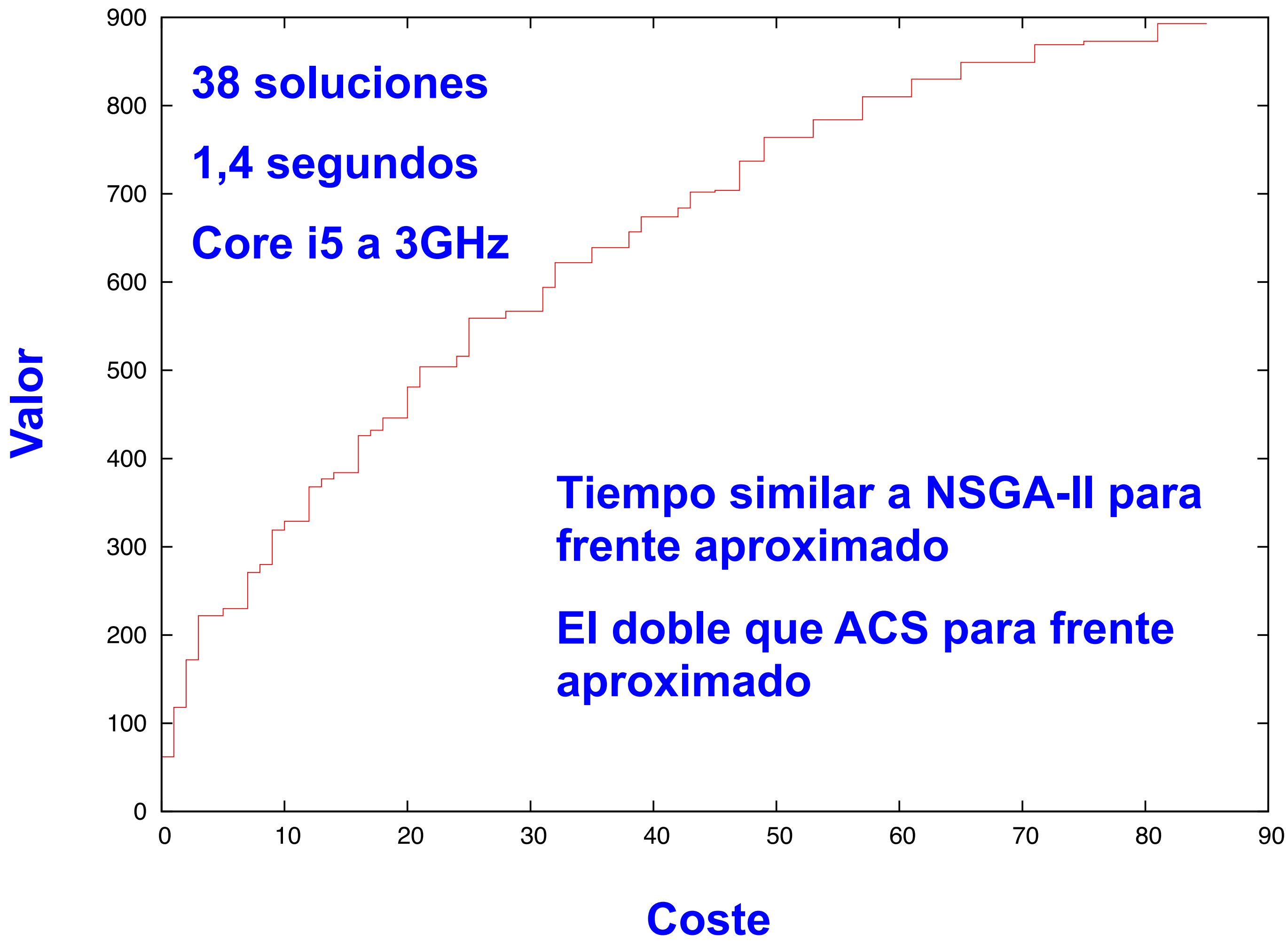
$$\text{máx} : \sum_{j=1}^{n} s_j r_j \quad \text{sujeto a:} \quad \sum_{j=1}^{n} c_j r_j \leq B.$$

ε-constraint para resolver NRP bi-objetivo



Veerapen, Ochoa, Harman & Burke, IST 65: 1-13

Resultados: instancia con 20 requisitos



JISBD
XX Jornadas de Ingeniería del Software y Bases de Datos

Resultados: instancia con 100 requisitos

Conclusiones y Trabajo Futuro

Conclusiones

- Los resolutores de SAT actuales son suficientemente potentes para resolver instancias relativamente pequeñas de problemas de optimización combinatorios
- Usando ε-constraint se resuelven problemas multi-objetivo

Trabajo Futuro

- Aplicar enfoque a otros problemas
- Explorar ILP

Gracias por su atención !!!

Agencia de Obra Pública de la Junta de Andalucía
CONSEJERÍA DE FOMENTO Y VIVIENDA

OTRI # 8.06/5.47.4356, AOP GGI3003IDII (MAXCT)

http://maxct.lcc.uma.es

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

TIN2014-57341-R (MoveOn)

Andalucía Tech
Campus de Excelencia Internacional

UNIVERSIDAD DE MÁLAGA

VŠB TECHNICKÁ UNIVERZITA OSTRAVA 1849