

Plataforma de resolució de problemes RCPSP utilitzant SMT

David Moreno Oliveras

4 de juny de 2013

Resum

El problema de la seqüenciació de projectes amb recursos limitats, *resource constrained project scheduling problem* (RCPSP) està considerat com el problema bàsic més important dins la seqüenciació amb recursos limitats i ha estat objecte d'estudi i investigació. El problema de trobar l'òptim en el RCPSP pertany a la classe de problemes NP-Complet. Per resoldre aquest problema es podria intentar a través de diferents teories de programació: SAT, LP, algorismes de Branch and Bound, etc. En aquest projecte, s'ha triat un solver SMT anomenat Yices amb la teoria de fons aritmètica lineal entera per resoldre el problema.

1 Introducció

El problema d'operacions (scheduling) és un procés de presa de decisions que juga un paper molt important en organitzacions de manufactura i serveis, ja que té una aplicació a la producció, transport i distribució, i a la comunicació i processament d'informació, entre d'altres. Consisteix en assignar d'una manera apropiada els recursos disponibles per al processament de tasques de manera que es puguin optimitzar els objectius de la organització.

Com cas particular de la programació d'operacions, hi ha la programació de projectes (*Project Scheduling*), que és el procés de planificar, organitzar i controlar activitats i recursos per aconseguir un objectiu concret, generalment amb limitacions de temps, recursos o costos. Dins aquest grup es situen els problemes de programació de projectes (PSP), que és un nom genèric que es dona a tota una classe de problemes en els quals és necessària la programació de manera òptima el temps, el cost i els recursos dels projectes.

Els paràmetres generals del PSP són les activitats i els recursos. Les varis tipus d'activitats i aquestes poden ser preemptives i no-preemptives, les primeres es basen en el fet que es poden executar segons les prioritats

del sistema, en l'instant de temps que calgui dependent de les necessitats o prioritats del sistema. En el cas de les activitats no-preemptives, en canvi, quan una activitat entra en execució, no pot parar d'executar-se fins que acabi. Hi ha varis tipus de recursos però principalment poden classificar-se en renovables i no renovables. En el cas dels renovables la disponibilitat del recurs està limitada a cada unitat de temps, en el cas dels recursos no renovables, estan limitats sobre la durada completa del projecte, de manera que no hi ha cap restricció en cada període.

Hi ha moltes variants del PSP, i existeixen moltes maneres de tractar aquests problemes però principalment existeixen dues estratègies per resoldre'ls: els algorismes analítics que garantitzen la solució òptima i els algorismes heurístics que, tot i que no garantitzen la solució òptima, sí produeixen solucions aproximades a l'òptim, en la majoria dels casos, en temps considerablement menors.

Aquest projecte, té com a objectiu mostrar i comparar els resultats obtinguts en l'estudi d'un dels problemes PSP: *el Resource-Constrained Project Scheduling Problem*, RCPSP. En el cas del RCPSP, les activitats són del tipus no-preemptives i els recursos únicament no renovables. Es pretèn trobar la solució òptima utilitzant mètodes analítics.

El problema de trobar l'òptim en el RCPSP és de naturalesa combinatoria i pertany a la classe de problemes NP-Completo, per la qual cosa, l'espai de solucions possibles creix de forma exponencial a mesura que s'incrementa el tamany del problema. La resolució d'aquests problemes demana molta velocitat de procés. Per tal de trobar una solució òptima, en aquest projecte s'ha triat resoldre el problema mitjançant un solver o resolador SMT (*satisfiability modulo theories*).

La *Satisfactibilitat Mòdul Teories* (SMT) és el problema de decidir la satisfactibilitat d'una fórmula de primer ordre sobre una teoria de fons de primer ordre. Una instància SMT és una fórmula de primer ordre on els símbols dels predicats ténen interpretacions predefinides d'acord amb una teoria de fons. Alguns exemples són la igualtat i funcions no interpretades, aritmètica lineal d'enters (LIA : *Linear Integer Arithmetic*), aritmètica lineal de reals, arrays, etc. Un problema SMT consisteix en determinar la satisfactibilitat de fórmules lògiques sense quantificadors, respecte a teories de fons expressades en lògica clàssica de primer ordre amb igualtat.

En el cas particular d'aquest projecte, la teoria de fons és la teoria aritmètica lineal entera, i s'ha triat com a mètode de resolució un solver o resolador SMT ja que és un mètode complet i això assegura que de les crides successives que es faran per trobar la solució òptima, si existeix la solució, es trobarà. I com a solver SMT s'ha escollit el Yices ja que és un resolador molt eficient i que s'ha demostrat que dóna un bon rendiment.

Així doncs, en aquest projecte es transforma el problema d'optimització

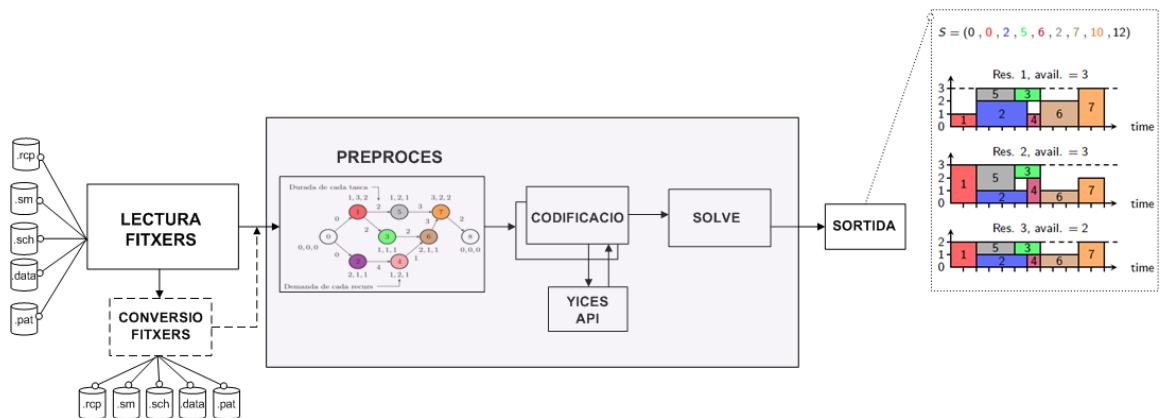


Figura 1: Arquitectura del sistema

de RCPSP en una fórmula lògica, on alguns dels àtoms de les clàusules d'aquesta fórmula pertanyen a la teoria LIA i després, s'intenta resoldre aquesta fórmula mitjançant el solver SMT triat, el Yices.

2 Objectius

La finalitat d'aquest projecte és crear una plataforma RCPSP que pugui llegir diferents formats d'entrada (fitxers del tipus `.rcp`, `.sch`, `.sm`, `.data`, `.pat`), pre-processar-los, codificar-los a través de diferents modelitzacions (TaskRD, TimeRD...) per tal de poder-los passar a instàncies SMT i poder executar-los a través de la API de Yices. L'objectiu és trobar el temps d'inici per cada activitat de manera que es minimitzi la longitud del makespan sense que es violin les restriccions.

Cal dissenyar una aplicació en C++, que sigui escalable i que pugui aconseguir el resultat del problema en el temps més òptim possible.

3 Arquitectura del sistema

L'arquitectura del sistema està composta per diferents mòduls. La figura 1 mostra els mòduls d'aquesta arquitectura. Inicialment hi ha la **lectura de dades** que es troben en un sol fitxer d'entrada, el qual pot estar en 5 tipus formats diferents (`.rcp`, `.sm`, `.sch`, `.data`, `.pat`). Un cop llegides les dades, en el cas que es vulgui, es pot **convertir les dades** a un fitxer amb el format de fitxer que s'hagi triat.

Posteriorment, hi ha una fase de **preprocés**, on es calcula el conjunt de precedències entre tasques (grafi del problema i matrius d'adjacències i precedències), altres càlculs com el Upperbound del makespan o el Timewindows per a cada activitat així com la matriu d'incompatibilitats entre les

activitats.

A continuació, a la fase de **codificació** es representa el problema segons una codificació : Time o Task. En aquesta fase, es tria una de les dues codificacions i es programen les restriccions de la codificació utilitzant la **API de Yices** (llibreria) per cridar el resolador perquè ens doni una solució(**solve**). Com que volem la solució òptima, es va cridant iterativament al resolador acotant la variable objectiu mitjançant una cerca dicotòmica per tal de trobar la solució òptima.

4 Conclusions

Com a conclusions es pot dir que s'han assolit els objectius inicials amb èxit:

- S'ha creat una plataforma que resol problemes RCPSP utilitzant SMT
- Donats un fitxer amb els paràmetres del problema, es troba el temps d'inici per cada activitat sense violar cap restricció i aconseguint minimitzar la longitud del makespan
- S'ha aconseguit que la plataforma fos modular de manera que si en futur es vol treballar amb algun altre solver que no sigui Yices, només s'haurà de modificar les funcions que criden a la API del nou solver, d'aquesta manera no s'haurà de programar de nou tota la fase de codificació i resolució
- Tot i que no estava en els objectius inicials, s'ha creat una implementació per convertir els fitxers d'entrada en fitxers amb el format desitjat dins els acceptats per el problema. D'aquesta manera s'ha aconseguit més agilitat amb els testejos i jocs de prova a l'hora d'executar el problema
- En quant als resultats obtinguts per l'aplicació cal destacar que sembla que la codificació Time és més constant i escalable amb el resultat de les execucions del problema i que tot i ser més lenta en resoldre la majoria instàncies, ha trobat l'òptim per totes les instàncies
- Tot i que no es pot observar en el joc de proves *j30*, per altres proves puntuals que s'han realitzat, sembla que la codificació Time seria millor per a instàncies on l'horitzó de temps és més petit ja que el nombre de variables creix amb l'horitzó de temps, és a dir, quan hi ha moltes activitats, el nombre de variables serà més gran i, en principi, hauria de costar més resoldre el problema. Donat que la codificació Task és independent de l'horitzó de temps semblaria que en instàncies on l'horitzó de temps és molt gran podria funcionar millor que la codificació Time.

5 Trebal futur

Hi ha determinades millores que es podrien tenir en compte a l'hora d'ampliar el projecte.

En primer lloc, es podrien desenvolupar algorismes més òptims amb càlculs més acotats en les codificacions desenvolupades. La millora dels càlculs com l'upper bound pot permetre millorar el temps d'execució i acotaria molt més la cerca de la solució òptima. En el cas del time window, si es pogués acotar al màxim possible, es podria reduir encara més el domini de cada variable i , per tant, millorar encara més la cerca.

En segon lloc, es podria intentar desenvolupar noves codificacions per tal de veure si es poden obtenir millors resultats.