

Paral·lelització de la simulació de la propagació d'incendis forestals

Albert Coca Abelló¹

Abstract—Els incendis forestals són un tipus de desastres naturals que cada any provoquen importants pèrdues a nivell mundial. Per a poder lluitar de la manera més eficient contra aquests desastres és de gran ajuda poder predir el comportament i evolució d'aquests incendis. Així, s'han desenvolupat diversos models de propagació i simuladors que intenten predir el comportament dels incendis. En aquest context, cal tenir en compte que el temps de simulació és un factor clau, ja que la predicció s'ha de realitzar molt més depressa que el temps real, per a poder prendre accions que mitiguin l'efecte dels incendis. En aquest treball s'ha analitzat el comportament i temps d'execució d'un simulador, àmpliament utilitzat en el camp, anomenat FARSITE, que presenta un temps d'execució força irregular. Un cop analitzat el simulador s'ha dut a terme una paral·lelització basada en pas de missatges (MPI) que ha permès reduir el temps de simulació de forma significativa.

I. INTRODUCCIÓ

En els últims anys hem vist que l'home ha causat intencionadament o de forma indirecta importants incendis forestals. Durant els últims 5 anys s'han declarat 2871 incendis només a Catalunya. També degut al canvi climàtic, que portarà temperatures més elevades i períodes de sequera més accentuats, farà que el nombre i la perillositat dels incendis s'incrementi. En la taula I es mostra el nombre d'incendis que s'han declarat a Catalunya, classificats segons les seves causes, així com l'àrea total cremada cada any. Cal remarcar el cas de l'any 2012 en el que els 747 incendis declarats van cremar un total de 15.025 hectàrees. A escala mundial, s'estima que cada any es cremen de mitjana uns 350 milions d'hectàrees.

TABLE I
CAUSES DELS INCENDIS FORESTALS A CATALUNYA

	2010	2011	2012	2013	2014
Accidentals	61	65	73	83	39
Intencionats	121	150	210	154	96
Naturals	32	41	44	76	83
Negligències	197	259	305	217	184
Reactivats	3	5	6	5	4
Desconegut	61	66	109	68	55
Total	475	586	747	602	461
Total ha	618	1097	15026	1059	1570

Analitzant la Taula I podem observar que els incendis forestals causen un gran impacte en el nostre mediambient i que cada any es cremen un gran nombre d'hectàrees. Per altra banda podem observar que més del 70% dels incendis són causats per l'acció humana i que gairebé una quarta part han estat provocats intencionadament i quasi el 40% ha estat per

¹Gràcies al Departament d'Arquitectura de Computadors i Sistemes Operariis de la UAB, i en especial al grup de recerca HPCA4SE.

la inconsciència de la gent, ja sigui per culpa de cigarretes, crema agrícola, fogueres o treballs forestals entre altres.

Cada cop s'han anat millorant més les tècniques per a apagar aquests incendis i per a poder prevenir-los. També s'han promulgat noves lleis per a castigar a la gent que causa algunes de les negligències, però ara estem a l'era de les noves tecnologies i com no podia ser d'una altra forma s'han introduït en aquest àmbit també.

Una eina que resulta molt útil per a lluitar contra els incendis i fer servir els recursos disponibles de la millor manera possible durant l'extinció és la predicció de la propagació dels incendis. Per això s'han desenvolupat models de propagació que donats uns paràmetres d'entrada proporcionen una aproximació de l'evolució de l'incendi de les properes hores. Hi ha models de diferents característiques, però podem dir que, en general, reben un conjunt de paràmetres d'entrada que inclou la topografia del terreny, el mapa i les dades de la vegetació, la informació meteorològica i l'estat inicial de l'incendi. Amb aquestes dades, els models avaluen la velocitat, direcció i intensitat de l'incendi en els propers instants de temps.

Aquests models s'han incorporat en eines de simulació que permeten obtenir els resultats molt més de pressa del que es faria si el model es resolgués de forma manual. Per aquest motiu s'han vist la necessitat d'utilitzar la computació per ajudar a les tasques d'extinció d'incendis. El fet de poder preveure cap a quina direcció es propagarà un incendi permet poder utilitzar els recursos físics d'una forma més eficient i, per tant, mitigar els efectes dels incendis forestals i reduir el cost mediambiental que suposen.

Diversos grups d'investigació han creat simuladors per a intentar ajudar a les tasques d'extinció d'incendis. Alguns dels simuladors existents són: FARSITE [7], WildFire Analyst [8], FireStation [9] i FIRETEC [10].

Un estudi que avalua aquests simuladors [1], entre altres, conclou el següent: "The comparison discussion concludes with the FARSITE simulator model which is the one that stands out from the rest". És a dir, FARSITE és el millor simulador dins de l'estudi. Per això hem escollit aquest simulador per a fer les nostres prediccions d'incendi i hem analitzat el seu funcionament i rendiment, per a intentar millorar les prestacions del simulador.

II. EL SIMULADOR FARSITE

FARSITE és un simulador de propagació d'incendis forestals que, utilitzant dades d'entrada de les condicions mediambientals de la zona del incendi, intenta predir quina serà la propagació de l'incendi al cap d'unes hores o dies. Per fer-ho, el simulador necessita un perímetre inicial de

foc i, mitjançant les equacions de propagació, obté un nou perímetre de sortida que equivaldrà a la predicció de l'incendi. En la Figura 1 es mostra un exemple de propagació d'un incendi, indicant els punts més importants amb la nomenclatura utilitzada a FARSITE.

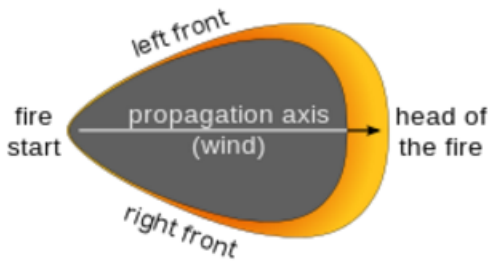


Fig. 1. Propagació del foc

FARSITE utilitza diversos arxius d'entrada per a poder fer la simulació, principalment tenim el "landscape" (LCP) que conté l'elevació del terreny, la inclinació, el model de fuels i la densitat com a paràmetres mínims. Però també podem afegir-ne d'altres com el tipus de fusta, l'alçada dels arbres, l'alçada del sotabosc i la seva densitat. A més del LCP necessitem un mapa d'humitats per a poder iniciar la simulació. El programa també permet afegir a la simulació un model de fuels personalitzat, un model del clima i com no podia faltar, un model de vents. L'altra dada d'entrada necessària és el front de l'incendi en l'instant inicial. Aquest front ve donat per un conjunt de punts ordenats que determinen el perímetre de l'incendi. Amb aquest conjunt d'entrades FARSITE proporciona el nou front de l'incendi al cap d'un cert interval de temps. En la figura 2 podem veure els diferents paràmetres d'entrada del simulador on els paràmetres de color vermell són necessaris i els de color negre opcionals.

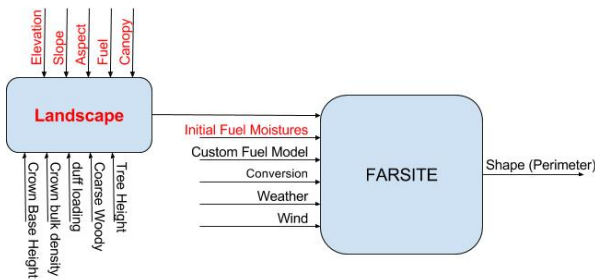


Fig. 2. Entrades i sortides del simulador FARSITE

Per a determinar la propagació de l'incendi, FARSITE agafa el perímetre inicial i a cada punt aplica les equacions de propagació del foc per a calcular la velocitat de propagació a cada punt i així determinar la nova posició de cada punt al cap d'un cert interval de temps (*TimeStep*). És a dir, s'aplica un principi de propagació semblant al principi de Huygens. Un cop que s'han obtingut els nous punts del front, FARSITE agrega els punts per a conformar el nou

front. Aquest nou front serveix com a entrada del proper pas de simulació, i així va evolucionant el simulador. Així, FARSITE ens donarà, per a cada simulació, com s'ha anat propagant el foc, i així, podem saber quin camí ha seguit el foc. La figura 3 mostra com el simulador fa la propagació dels punts i com l'incendi s'expandeix.

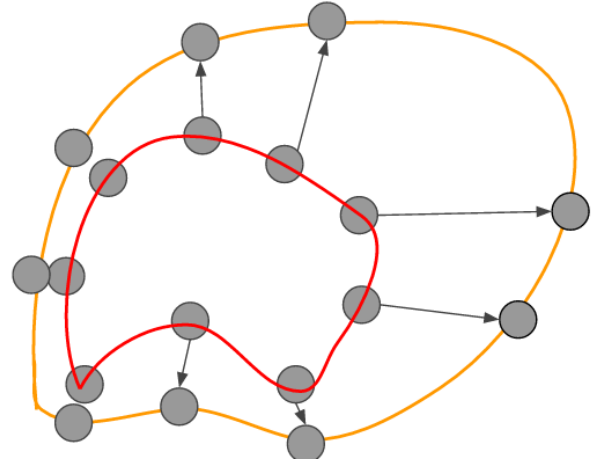


Fig. 3. Expansió del perímetre per punts

Per al funcionament del simulador hem de definir el pas de temps que representa cada pas de simulació. Com més petit sigui el pas de simulació, més acurada hauria de ser la simulació, però també implicarà una major quantitat de còmput. Però, a més a més d'aquest paràmetre hi ha una sèrie de paràmetres configurables per a cada simulació. Em centraré a explicar aquells que són més rellevants per a aquest treball:

- **Perimeter Resolution (PR):** és la distància màxima que hi pot haver entre dos punts consecutius en qualsevol perímetre del simulador en qualsevol moment. Això implica que si el front de l'incendi creix i els punts que el defineixen s'allunyen, és necessari incorporar nous punts per a mantenir condicions de continuïtat. FARSITE aplica mètodes d'interpolació per a agregar nous punts al front quan es produeix aquesta circumstància.
- **Visible Time Step (VS):** determina cada quants passos de simulació, el simulador guarda un perímetre per a que puguem visualitzar-lo al finalitzar de la simulació.
- **Distance Resolution (DS):** un cop que tenim definit el *TimeStep* s'aplica a cada punt les equacions de propagació per a determinar la velocitat de propagació i, així, obtenir el moviment de cada punt del front. Però, si el *TimeStep* és llarg i la velocitat de propagació resulta ser elevada, llavors els punts recorren molta distància i és possible que en aquest recorregut les condicions hagin canviat, fent la predicció poc acurada. Per evitar aquest problema, es defineix un nou paràmetre anomenat *DistanceResolution* que determina la propagació màxima de cada punt, de manera que si la distància recorreguda en el temps *TimeStep* és més gran que aquest valor, es limita el temps d'aquest pas al temps

necessari per a recórrer aquesta distància màxima. Aquest paràmetres de configuració es mostren a la Figura 4.

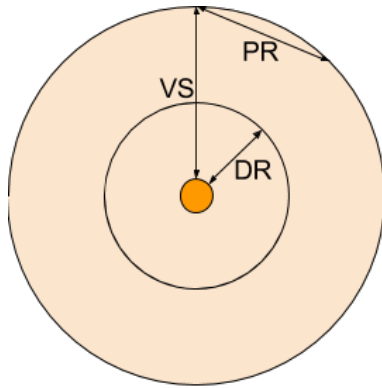


Fig. 4. Paràmetres de configuració

Amb els paràmetres anteriors podem controlar la qualitat de la simulació; com més petits siguin aquests paràmetres més precisió tindrem en les nostres simulacions. Això depèn també de la resolució dels paràmetres d'entrada, és a dir, del LCP.

III. ANÀLISI DE RENDIMENT

Després de realitzar diverses simulacions amb el FARSITE ens vam adonar que hi ha simulacions molt més lentes que altres, però això no vol dir necessàriament que els perímetres inicials d'aquestes simulacions siguin molt més grans.

Per això vam començar a analitzar els temps d'execució de les diferents simulacions. Mentre que en les simulacions ràpides, estàvem més temps fent els càlculs de la propagació d'incendis, en els casos més lents vam poder comprovar que no era així, i les funcions que utilitzaven més temps d'execució eren les funcions de correcció d'errors del perímetre. Com podem veure a la figura 5, les funcions que més temps d'execució utilitzen són funcions relacionades amb el creuament de punts.

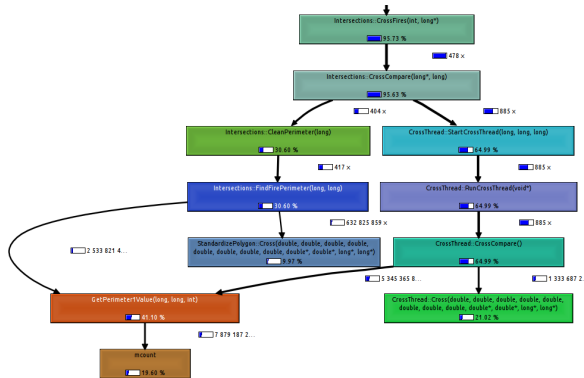


Fig. 5. Anàlisi del temps d'execució per funcions

En concret, el problema es dona quan la simulació no és tan perfecte com la de la Figura 3 i hi ha creuaments entre dos punts d'un perímetre, com podem veure en la 6. Això

fa que el perímetre no sigui consistent, per tan necessitem fer una serie de càlculs per comprovar el perímetre i per a corregir aquests errors. Això es el que alenteix el simulador.

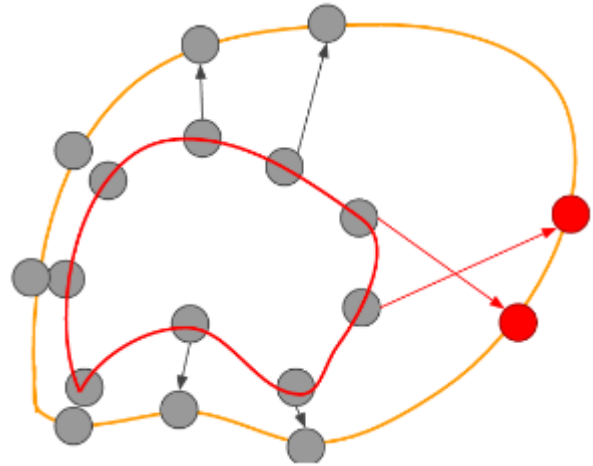


Fig. 6. Creuament de punts del front de l'incendi

També ens vam adonar que al incrementar el paràmetre DR i reduir el PR el temps de simulació augmentava considerablement. Això està directament relacionat amb el nombre de creuaments, ja que com més distància pot recórrer un punt (DR) i més aprop està del punt del costat (PR) llavors més possibilitats tenen aquests de creuar-se. Podem dir que augmentar la resolució dels punts augmenta el nombre de creuaments i que per tant la simulació va més lenta.

Podem dir doncs que una simulació sense creuaments, és a dir amb un perímetre regular, serà una simulació ràpida mentre que una simulació amb creuaments, és a dir amb un perímetre irregular, serà una simulació més lenta.

Els focs analitzats són simulacions en un terreny real dins del nostre país (Catalunya) buscant expressament casos en que el perímetre de l'incendi tingui formes poc regulars. Algunes coses que es van poder apreciar van ser que al reduir el PR el temps de simulació augmentava considerablement.

Com podem veure en la 5 la paral·lelització seria ideal en la funció intersections::CrossCompare ja que significaria aproximadament el 90% del temps del programa, però en aquesta funció el que fem és calcular cada pas de simulació; és a dir, necessitem haver calculat el pas de simulació anterior per a poder calcular el següent. Hi ha una dependència clara entre cada iteració del bucle principal d'aquesta funció.

Si seguim mirant la 5 podem veure que les funcions que depenen de la funció intersections::CrossCompare són intersections::CleanPerimeter amb un 30% del temps d'execució i la classe CrossCompare amb un 60% per aquest motiu hem decidit paral·lelitzar la segona, ja que utilitza més temps.

A. CrossCompare::Compare

Donats dos perímetres on el primer perímetre serà el perímetre inicial i el segon la propagació d'aquest perímetre, aquesta funció agafa cada un dels punts del perímetre inicial

i comprova si la propagació d'aquest punt es creua amb la propagació de qualsevol altre. Tal i com podem veure a la 7 es comprova el punt blau juntament amb la seva propagació amb la resta de punts per saber si es creuen o no. Així doncs, si hi ha alguna propagació que fa el perímetre inconsistent, perquè es creua amb algun altre punt, aquesta funció ho detecta. Cada cop que un creuament és detectat, aquesta funció crida la funció de resolució de conflictes de propagació. Aquesta funció és la funció `crossCompare:Cross`, que s'encarrega d'arreglar els creuaments.

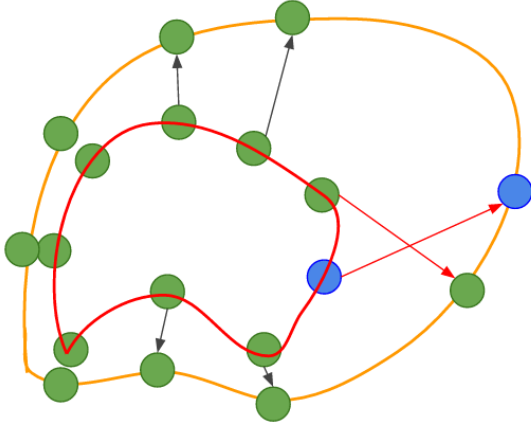


Fig. 7. Funció `CrossCompare`

B. `CrossCompare:Cross`

Donats dos punts creuats aquesta funció el que fa és trobar una forma de resoldre el creuament. Bàsicament ho fa utilitzant sentències condicionals que determinen quina de les diferents resolucions conegudes és millor per a cada cas.

Aquesta funció és la que més temps tarda durant la simulació. El que fa és fer els càlculs necessaris per a poder resoldre els conflictes de creuament de punts.

Quan el nombre de creuaments creix el temps de simulació augmenta dràsticament, aquest és el punt crític del simulador.

IV. PARAL·LELITZACIÓ DE FARSITE

S'han realitzat estudis d'optimització a l'hora d'obtenir dades per als incendis, com per exemple aconseguir models de vent complexes, que són molt importants per a una bona simulació en un temps reduït, però aquest treball no es centrarà en l'obtenció de dades sinó que ho farà en els següents aspectes:

A. Millorar el temps d'execució

El que pretén aquest treball és aconseguir que la previsió d'un incendi es faci el més ràpid possible i garantir que el temps d'execució mai sigui superior al de simulació podent ajudar als serveis d'extinció a tindre una previsió de l'expansió del foc ràpidament i que això els permeti actuar amb rapidesa i seguretat a l'hora d'atacar un incendi i reduir el nombre d'hectàrees cremades reduint la desolació en escenaris com el següent:

B. Afavorir els algorismes d'aprenentatge computacional

El treball també pretén afavorir la utilització d'algorismes d'aprenentatge computacional per a poder aconseguir una previsió més acurada a la realitat. Normalment aquests algorismes necessiten executar múltiples vegades la mateixa aplicació canviant els paràmetres i per tant, si aconseguim reduir el temps d'execució de l'aplicació podem millorar també el temps que triguen aquests algorismes en donar un resultat.

V. PARAL·LELITZACIÓ

Decidim dividir l'aplicació en N processos, cada un d'ells executarà una part de l'algorisme de correcció d'errors,

El treball que s'ha dut a terme és la paral·lelització d'aquesta funció dividint els punts entre N i fent que cada procés faci la comprovació de K/N punts, on N és el nombre de processos i K el nombre de punts.

Per aconseguir-ho cadascun dels N processos executa la funció de comparació i cada una d'elles calcula el nombre de creuaments que hi ha en el seu rang de punts i finalment es comuniquen tots els processos amb la resta per comunicar els creuaments que ha trobat cadascun d'ells.

S'ha utilitzat aquesta forma perquè així la quantitat de dades a enviar és menor, ja que només s'han d'enviar els punts que s'han creuat i no la seva totalitat.

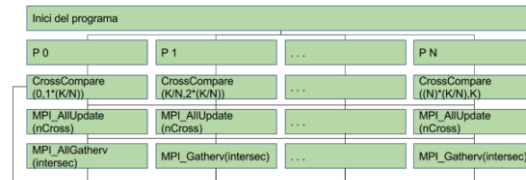


Fig. 8. Distribució del treball

Com podem veure en la Figura 8 després de la creació dels processos tots ells executen el mateix codi, cadascú amb un rang de punts diferent. Seguidament cada procés envia el nombre de punts creuats que ha trobat i al final envia els punts creuats (ids) a la resta de processos.

Així doncs crearem els processos MPI a l'iniciar la simulació i tots ells carregaran les dades necessàries i faran el primer pas de la simulació fins a arribar a les funcions de verificació del perímetre on cada procés MPI calcularà la seva part dels creuaments, com hem dit anteriorment cada procés MPI verificarà K/N punts del perímetre i s'encarregarà de solucionar els possibles creuaments.

VI. RESULTATS

Després de fer l'anàlisi inicial de l'aplicació vam determinar que la part paral·lelitzada és aproximadament un 60% del temps total de l'aplicació. Així, s'esperava aconseguir un speedup teòric de:

$$SpeedUp_{Teoric} = \frac{TS}{TS * 0.4 + (TS * 0.6)/NP}$$

Si analitzem la taula II podem observar dues coses: la primera és que sorprenentment per a les execucions amb 2 i 4 processos el speedup obtingut és més gran que el màxim teòric; la segona és que amb 16 threads el rendiment del programa comença a disminuir substancialment.

TABLE II
TAULA DE RESULTATS D'EXECUCIÓ

Processos	Temps	SpeedUp	SpeedUp Teòric
Serie	1:35:04	1,00	1,00
2	1:04:50	1,46	1,42
4	0:50:30	1,88	1,81
8	0:48:17	1,96	2,10
16	0:58:55	1,61	2,28

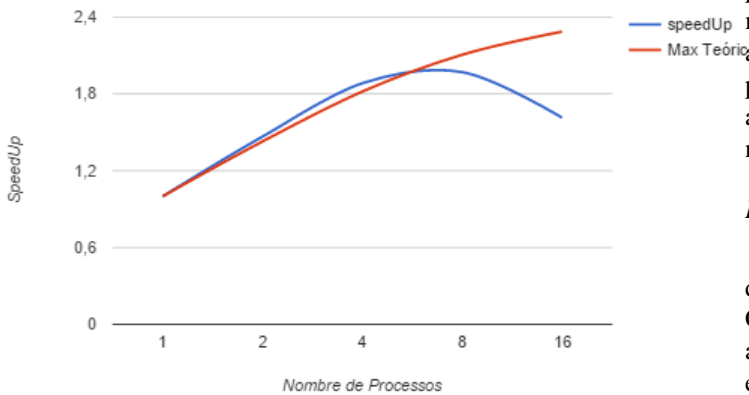


Fig. 9. SpeedUp

Si mirem els resultats de la taula II i la figura 9 podem veure que els resultats per a 2 i 4 processos són superiors al speedup teòric. Això és degut al fet que cada simulació pot tenir un nombre de creuaments diferent depenent dels paràmetres d'entrada i de les diferents condicions mediambientals. Com ja hem dit anteriorment com més creuaments hi ha en una simulació, més temps tarda aquesta i a la vegada pot augmentar la quantitat relativa als càlculs per a la correcció de punts. És a dir, la part paral·lelitzada. El temps de simulació no depèn del tamany del problema únicament sinó que també depèn del nombre de creuaments trobat. Per aquest motiu podem aconseguir superar el speedup màxim teòric.

Si analitzem els resultats per a 16 processos, podem veure que el speedup disminueix respecte als resultats obtinguts amb 8 processos, això és degut al nombre de comunicacions globals. En augmentar el nombre de processos cada un d'ells fa els càlculs per a menys punts i per tant es redueix el nombre de càlculs per a cada procés i el treball que ha de dur a terme, per contra augmentem el nombre de comunicacions que han de fer els processos per comunicar les correccions realitzades a la resta.

També s'ha de mencionar que l'aplicació amb molts processos té un overhead massa gran i acaba anant més lenta del que s'esperaria que ho fes. Això és perquè cada procés

té menys treball a fer i ha d'enviar més missatges, per això pot ser que l'overhead augmenti substancialment quan hi ha més processos si no augmentem el nombre de creuaments.

VII. POSSIBLES MILLORES

Després d'analitzar el simulador podem veure que hi ha millores a fer en el propi simulador, algunes relacionades amb el treball realitzat i altres que no tant però que tot i així podem ajudar a millorar la velocitat del simulador. A continuació mostrem les possibles millores a realitzar a la nostra versió.

A. Assignació dinàmica de creuaments

Si assignéssim els punts creuats de forma dinàmica la càrrega per a cada procés seria més similar, per tant els processos no s'haurien d'esperar tant i l'overhead seria menor. Es podria implementar un master-worker per a dur a terme aquesta tasca, però en aquest cas el coll d'ampolla podria ser el màster perquè hauria d'enviar molta informació a la resta de processos, o bé la part de comparar els punts no seria paral·lela.

B. Heurística de Creuaments

Intentar crear una funció heurística que pugui determinar quins punts mai es creuaran optimitzant així la funció cross-Compare. Els punts que estiguin molt lluny mai s'arribaran a creuar per tant no necessitem fer la comprovació de què ho estan fent. D'aquesta manera reduiríem les operacions totals de l'aplicació i això també ajudaria a poder implementar la primera millora proposada, ja que reduiríem el temps dedicat a la comprovació dels punts.

C. Millora de la funció CrossCompare::Cross

Aquesta funció ara mateix no està gens optimitzada, ja que té moltes sentències condicionals que fan alentir el simulador perquè el predictor de salts de la CPU és incapaç de predir quina serà la següent instrucció a executar, reescriure la funció o simplificar-la per reduir el temps del programa sèrie, aquesta seria la millor forma d'optimitzar l'aplicació

VIII. CONCLUSIONS

En aquest treball s'ha dut a terme una anàlisi d'un simulador de la propagació d'incendis forestals, anomenat FARSITE, per tal de poder millorar el seu temps d'execució. S'han determinat aquelles funcions del simulador que consumien més temps de còmput i s'ha fet una primera paral·lelització amb pas de missatges (MPI) que ha millorat el temps d'execució de forma significativa, i assolint una millora molt propera a la millora teòrica esperada. Aquesta aproximació s'ha mostrat efectiva, però l'anàlisi dels temps d'execució aconseguits posen de manifest que és necessari fer un canvi en la concepció de l'algorisme utilitzat, ja que en el seu moment no es va dissenyar pensant en paral·lelitzar-lo i la versió aconseguida arrossega molts problemes del disseny original.

REFERENCES

- [1] Papadopoulos, G.D. "A Comparative Review on Wildfire Simulators". Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2011 IEEE International Conference on, 24 marzo 2011: 233 - 243.
- [2] Sanjuan, G. "Applying domain decomposition Schwarz method to accelerate wind field calculation". High Performance Computing & Simulation (HPCS), 2015 International Conference on, 20-24 July 2015: 484 - 490
- [3] Cencerrado, A. "On the Way of Applying Urgent Computing Solutions to Forest" Fire Propagation Prediction". International Conference on Computational Science, ICCS: 2012.
- [4] Brandon, M. Collins. "Fire weather and large fire potential in the northern Sierra Nevada". Agricultural and Forest Meteorology, Volumes 189-190, 1 June 2014: 30-35
- [5] Cencerrado, A. "Relieving Uncertainty in Forest Fire Spread Prediction by Exploiting Multicore Architectures". International Conference on Computational Science, Volume 51, 2015: 1752-1761.
- [6] Artés, T. "Relieving the Effects of Uncertainty in Forest Fire Spread Prediction by Hybrid MPI-OpenMP Parallel Strategies", International Conference on Computational Science, Volume 18, 2013: 2278-2287.
- [7] Finney, Mark A. "FARSITE: Fire Area Simulator-model development and evaluation", U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 2004: 47.
- [8] Ramírez, J, Monedero S. "New approaches in fire simulations analysis with Wildfire Analyst" The 5th International Wildland Fire Conference, Sun City, South Africa: 2011
- [9] Lopes A.M.G, Cruz M.G, Viegas D.X. "FireStation — an integrated software system for the numerical simulation of fire spread on complex topography" Environmental Modelling & Software Volume 17, Issue 3, 2002: 269–285.
- [10] Linn R, Reisner J, Colman J.J, Winterkamp J. "Studying wildfire behavior using FIRETEC" International Journal of Wildland Fire 11(4),2002: 233-246.