In this document we present the partial order graphs generated for the different executions of each project.

DATA SET: PURE **PROJECT:** Video Search

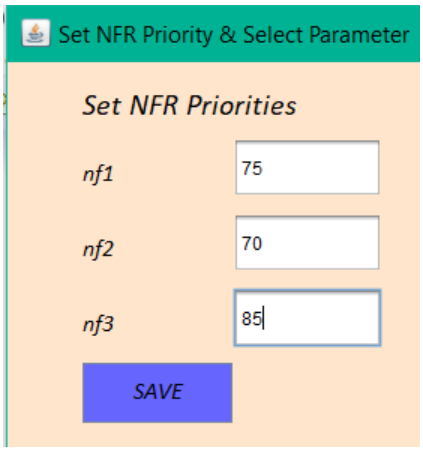

Figure 1: NFR Priority values assigned

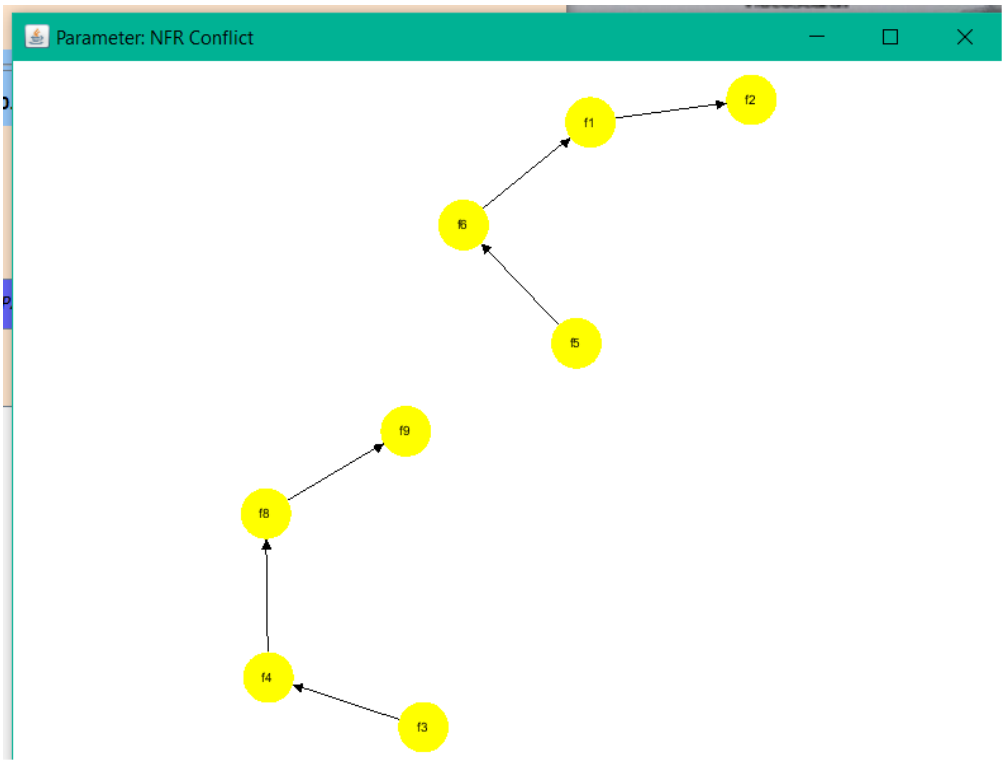

Figure 2: Partial Order when parameter is conflict

In this document we present the partial order graphs generated for the different executions of each project.

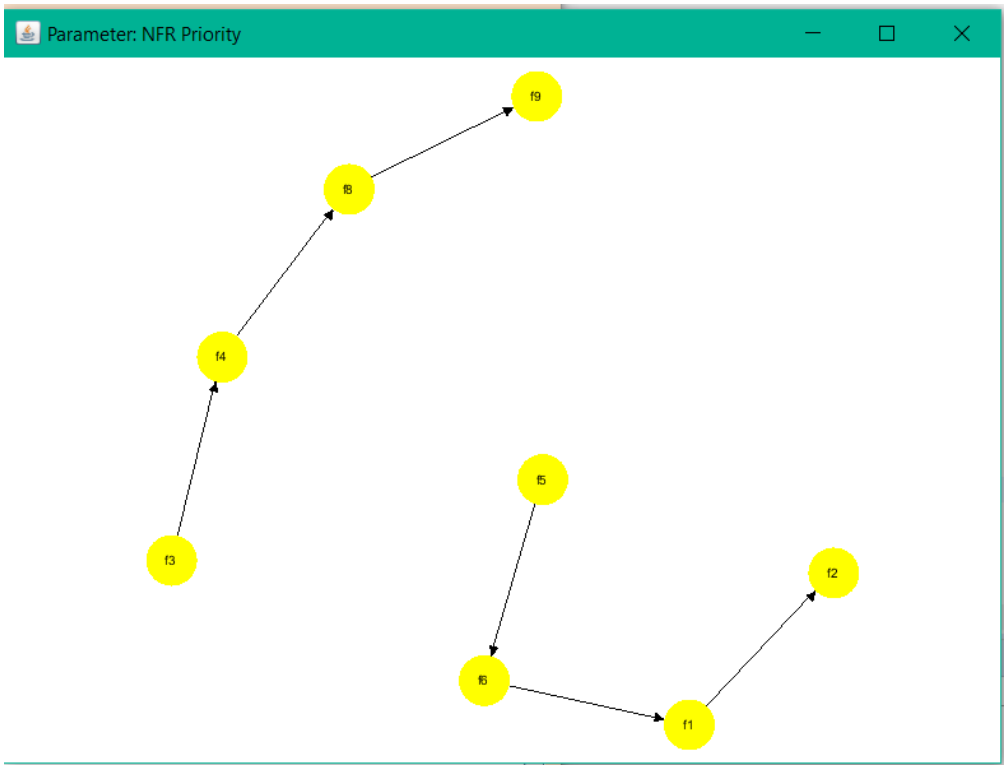

Figure 3: Partial order when parameter is NFR Priority

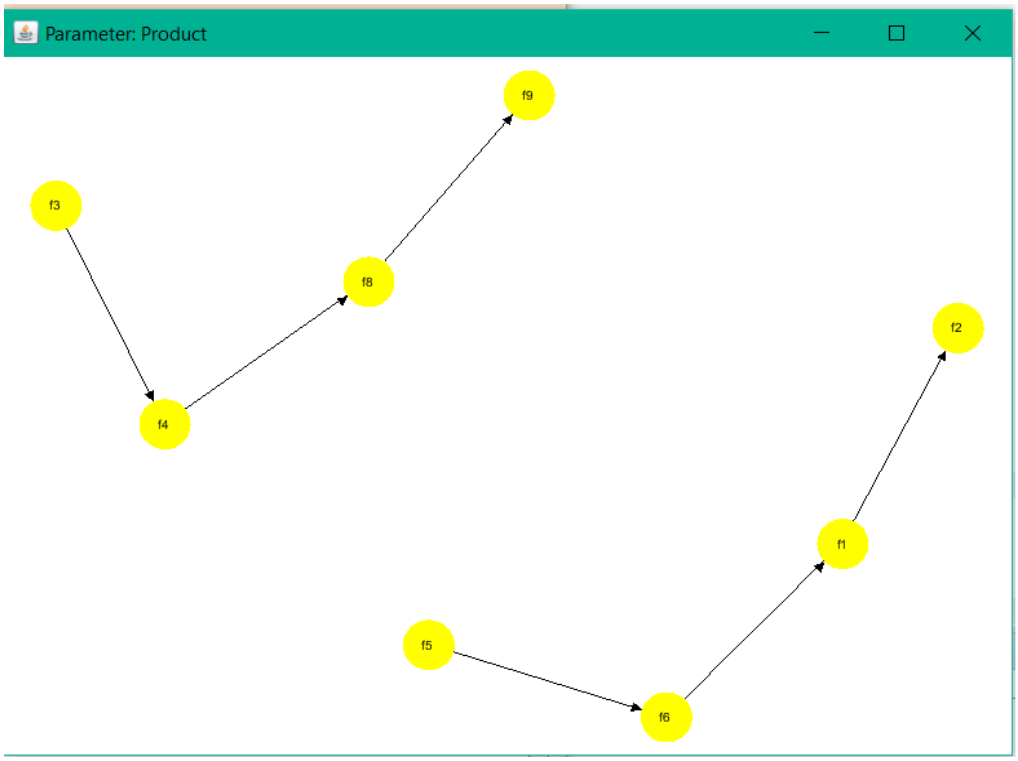

Figure 4: Partial order when parameter is product of NFR priority and conflict

In this document we present the partial order graphs generated for the different executions of each project.

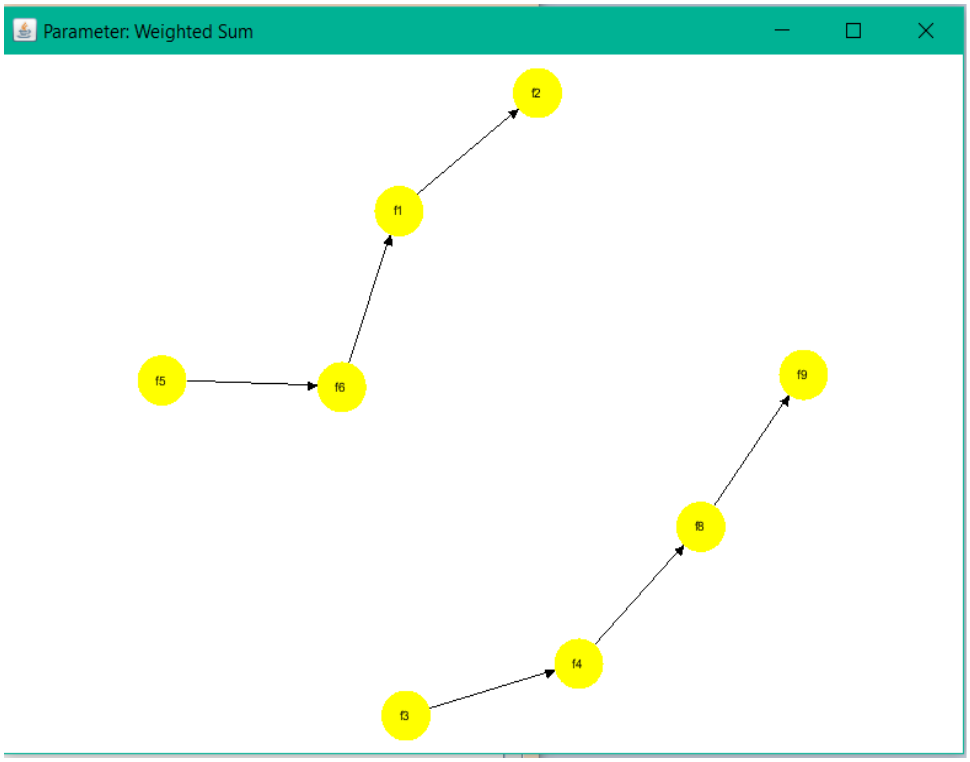

Figure 5: Partial order when parameter is weighted sum of NFR priority and conflict

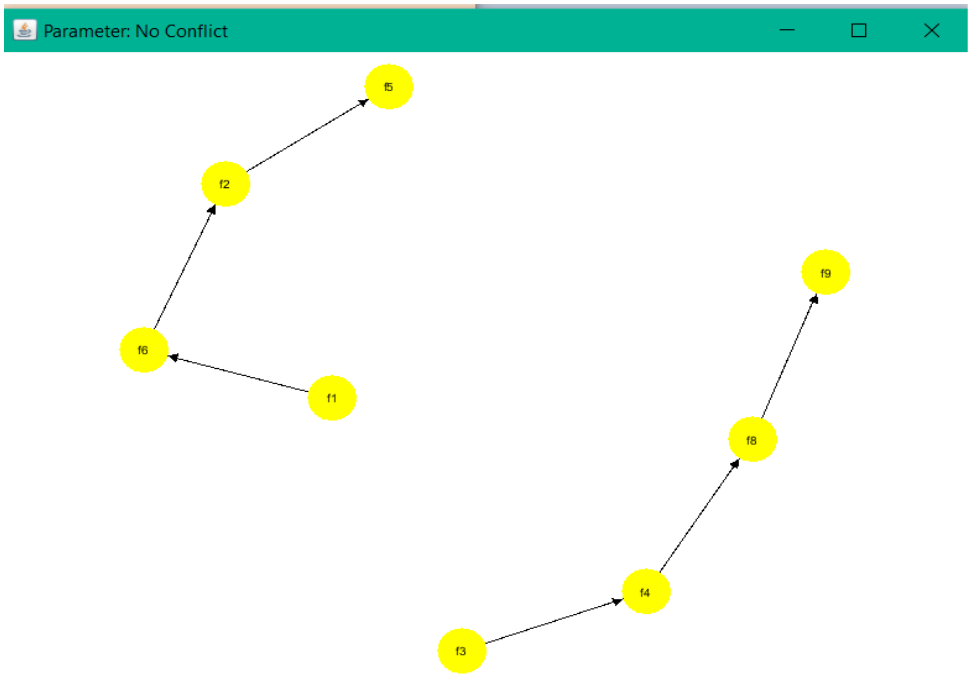

Figure 6: Partial order when conflict information is neglected

In this document we present the partial order graphs generated for the different executions of each project.

Summary:

The partial orders for all four parametric cases of our proposed approach are the same in this scenario (Figures 2, 3, 4 and 5). This is due to the fact that only a single macro cluster will be created during execution (as there is only one NFR conflict in the requirements set).

A partial order without considering the NFR conflict information is also generated (Figure 6). It can be observed that ordering among the FRs changes when NFR conflicts are ignored. This is because macro cluster information will not be considered and orders will be based on FR-NFR dependency in a basic cluster only. Of course, there will be some overlapping portions with the parametric cases (FRs f3, f4, f8, f9 has the same ordering in all the partial order graphs). Now if we consider the FRs f5 and f2 we find that there corresponding NFRs (performance & safety. Refer to requirements set) are in conflict. Hence in the ordering f5 precedes f2 in the first four parametric cases. However, without considering the conflicts the order changes to f5 follow f2, from the single basic cluster information. This change in the partial order may be helpful in analyzing the risks that may occur when functionalities are implemented disregarding NFR conflicts.

In this document we present the partial order graphs generated for the different executions of each project.

DATA SET: PURE **PROJECT:** Digital Home

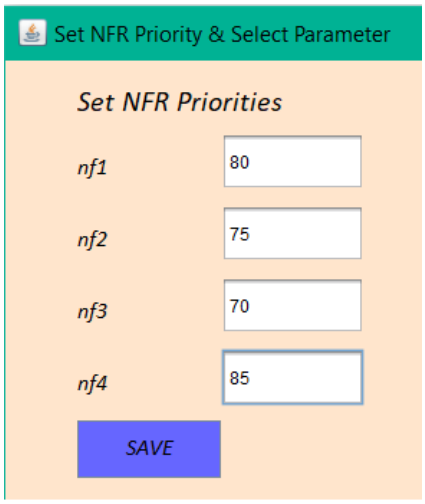

Figure 7: NFR Priority values assigned

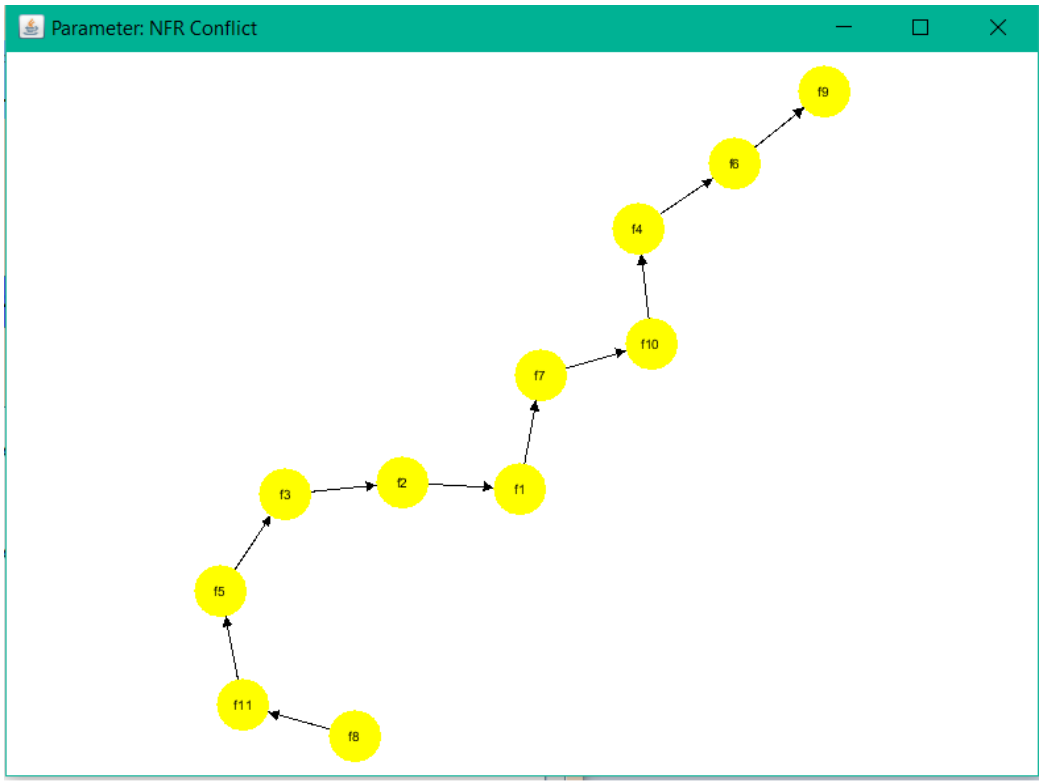

Figure 8: Partial order when parameter is NFR conflict

In this document we present the partial order graphs generated for the different executions of each project.

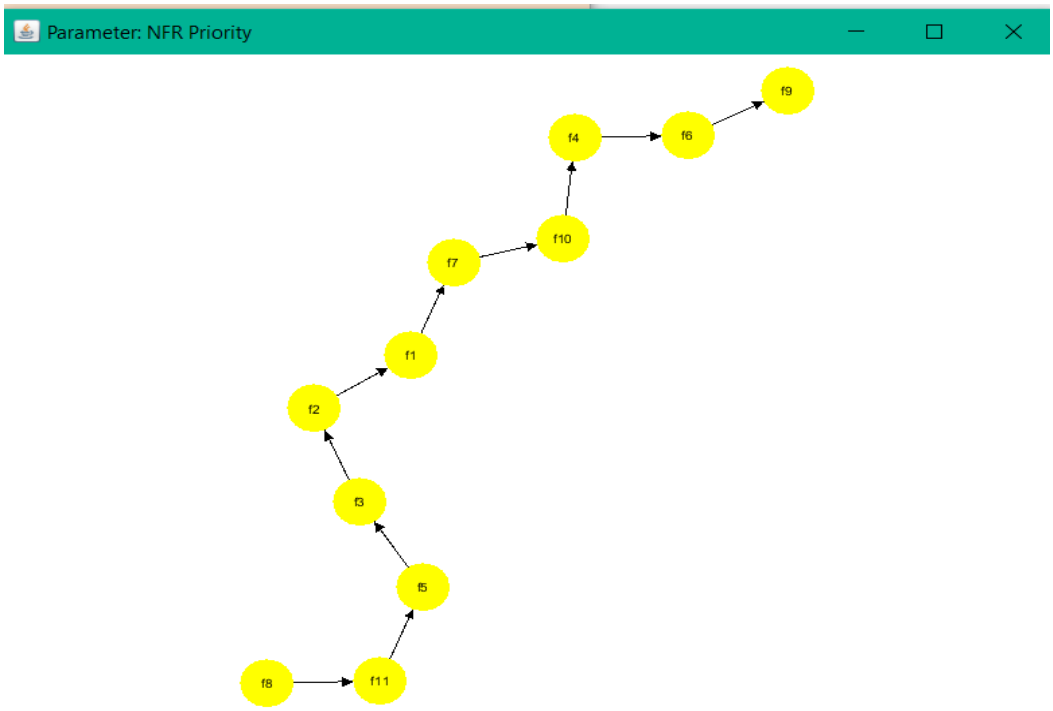

Figure 9: Partial order when parameter is NFR priority

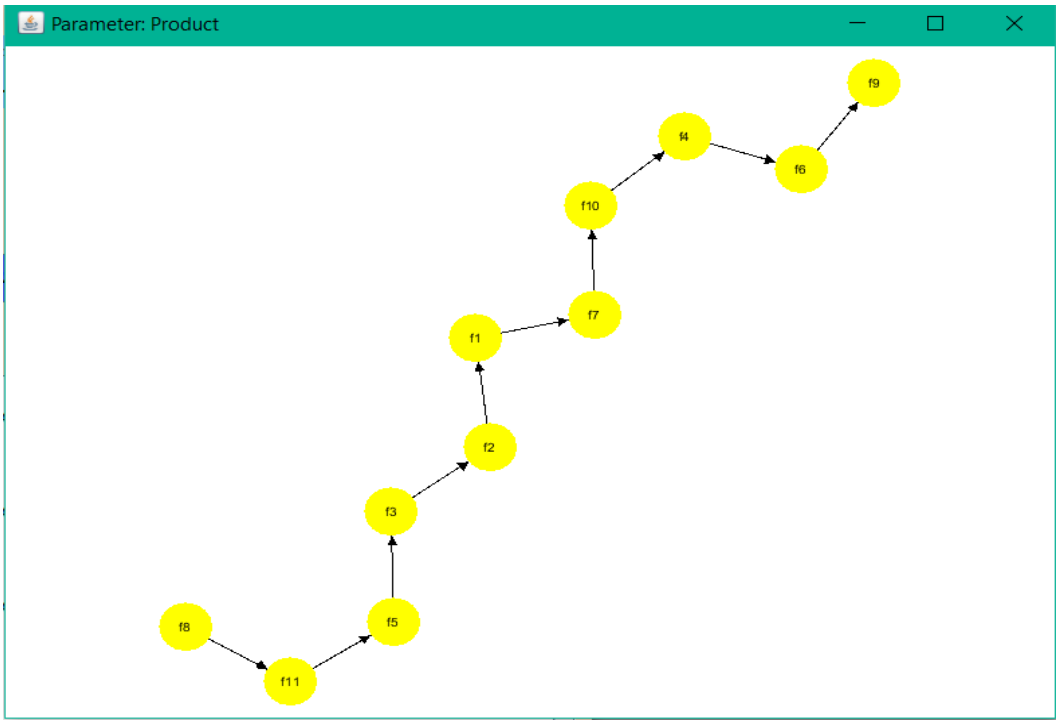

Figure 10: Partial order when parameter is product of NFR priority and conflict

In this document we present the partial order graphs generated for the different executions of each project.

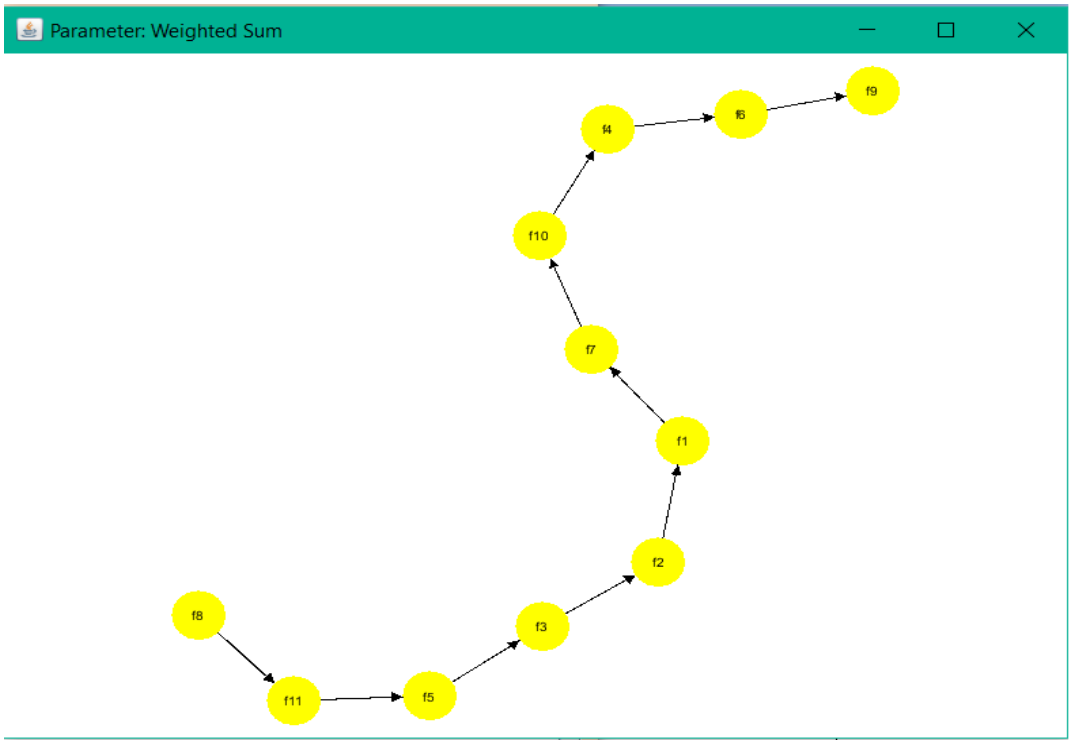

Figure 11: Partial order when parameter is weighted sum of NFR priority and conflict

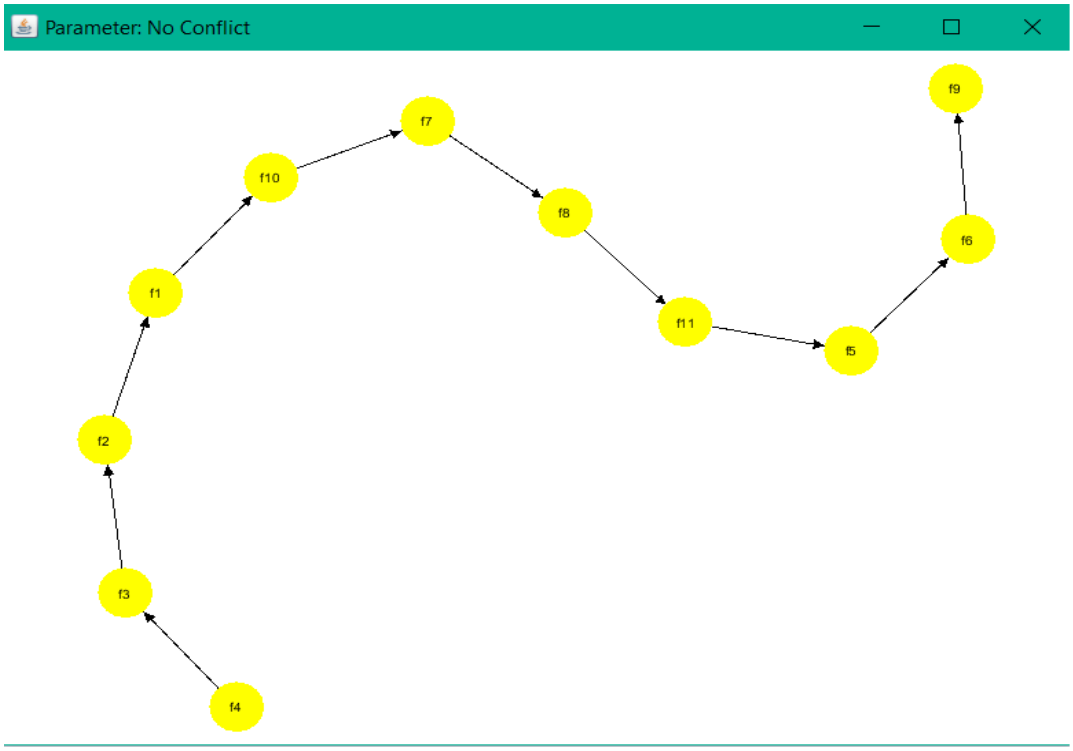

Figure 12: Partial order when conflict information is neglected

In this document we present the partial order graphs generated for the different executions of each project.

Summary:

The partial orders for all four parametric cases of our proposed approach are the same in this scenario (Figures 8, 9, 10 and 11). Although in this case we have two different macro clusters (for the two different NFR conflicts. Refer to requirements set), they share almost the same set of FRs. So, due to this and also because of conflict and priority values assigned the result remains unchanged.

A partial order without considering the NFR conflict information is also generated (Figure 12). It can be observed that ordering among the FRs changes **significantly** when NFR conflicts are ignored. This is because macro cluster information will not be considered and orders will be based on FR-NFR dependency in a basic cluster only. Of course, there will be some overlapping portions with the parametric cases (FRs f6, f9 has the same ordering in all the partial order graphs) due to NFR priority information alone. Let us consider the FRs f4 and f3. In all the parametric cases we observe that f3 precedes f4 due to conflict among their respective NFRs (performance and reliability. Refer to requirements set). However, without considering the conflict the order changes to f4 precede f3. This is because ordering is determined using only NFR priority and FR-NFR dependency values.

In this document we present the partial order graphs generated for the different executions of each project.

DATA SET: PROMISE **PROJECT ID:** 1

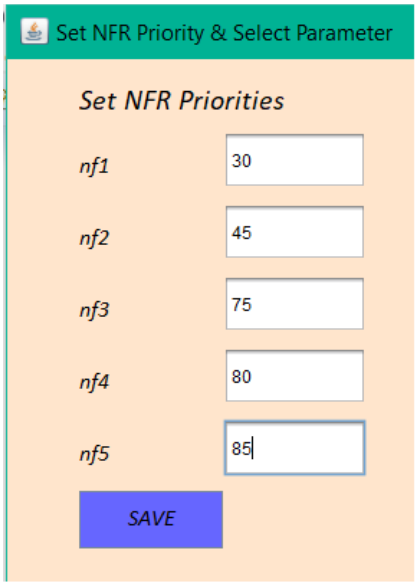

Figure 13: NFR priority values assigned

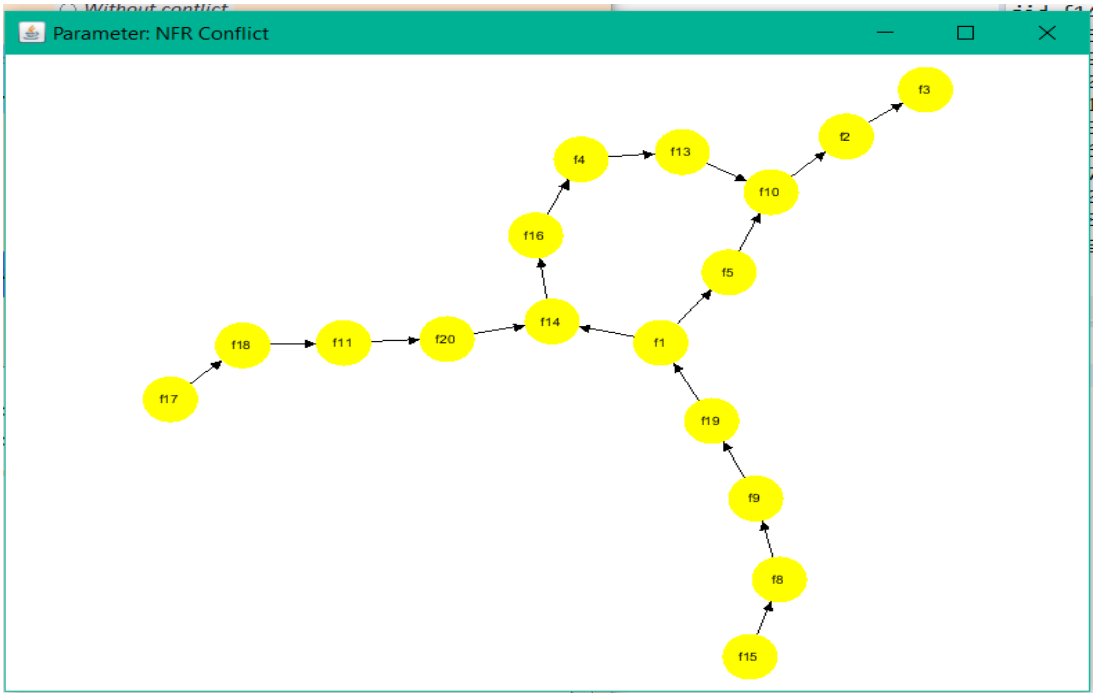

Figure 14: Partial order when parameter is NFR conflict

In this document we present the partial order graphs generated for the different executions of each project.

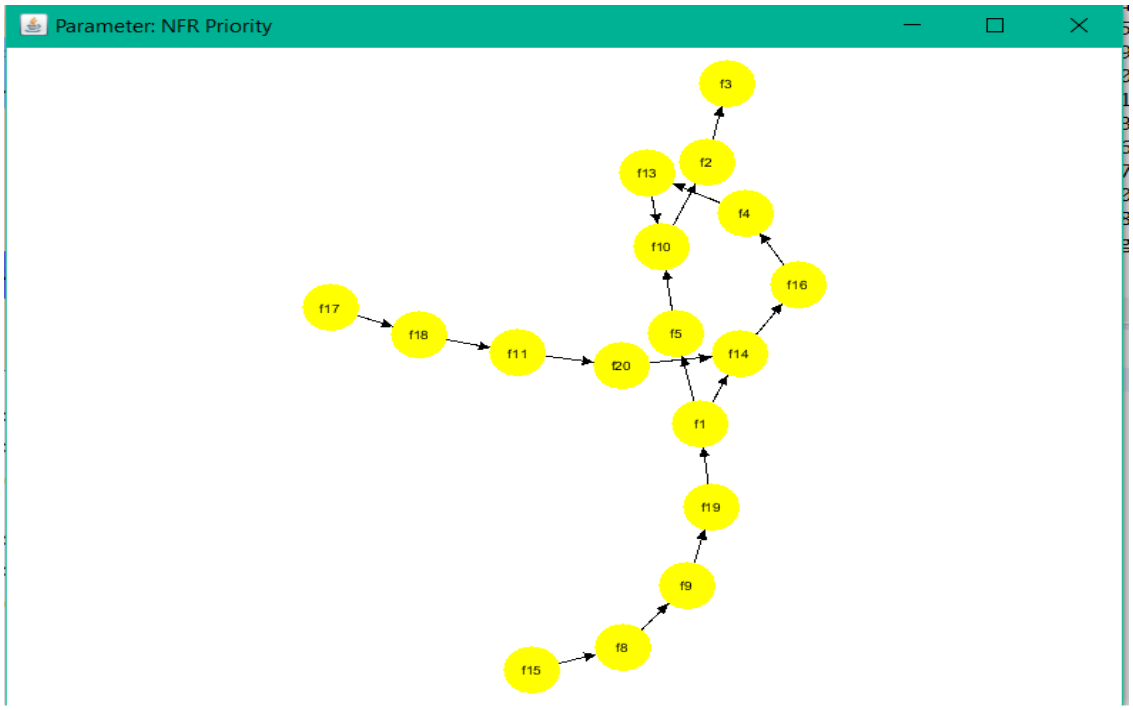

Figure 15: Partial order when parameter is NFR priority

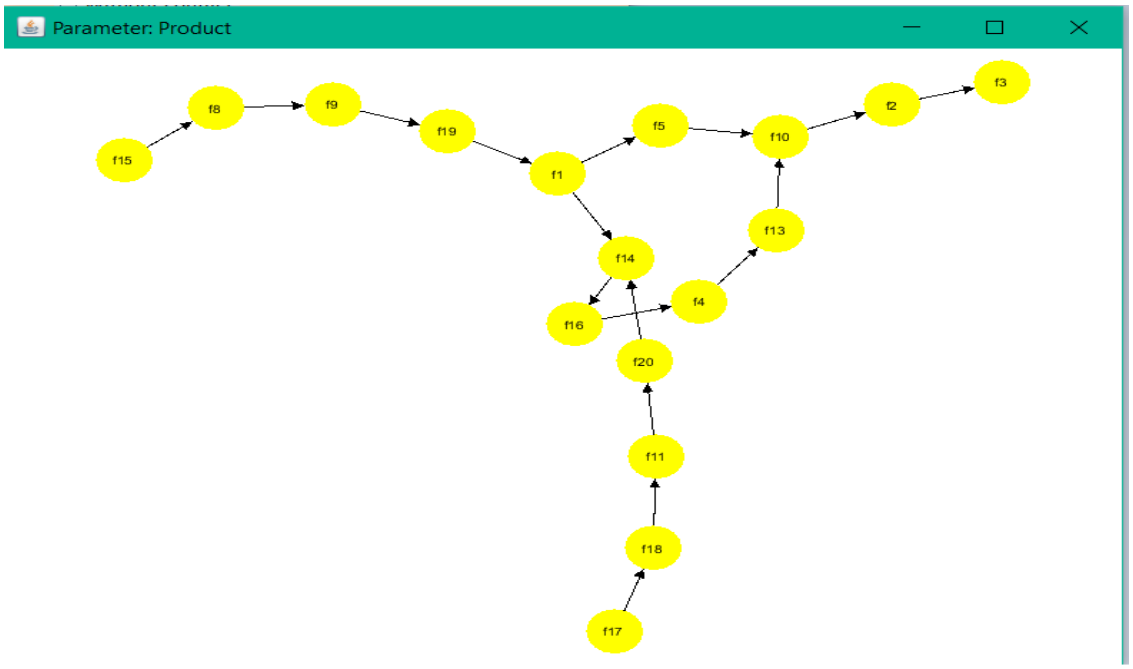

Figure 16: Partial order when parameter is product of NFR priority and conflict

In this document we present the partial order graphs generated for the different executions of each project.

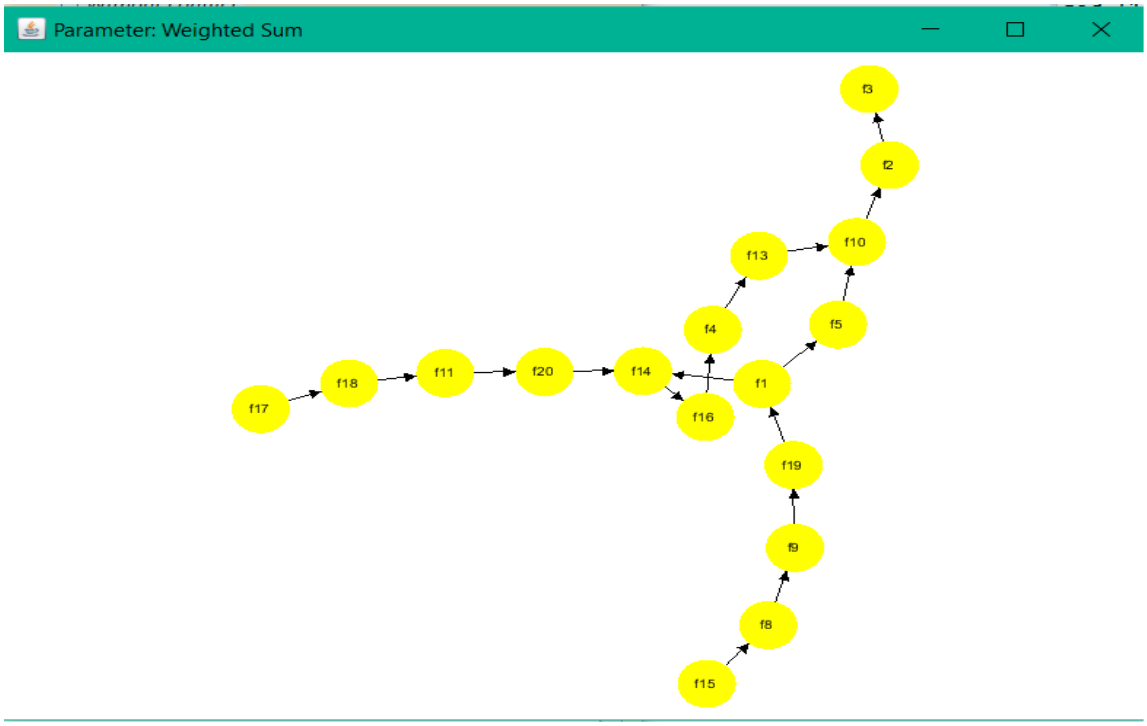

Figure 17: Partial order when parameter is weighted sum of NFR priority and conflict

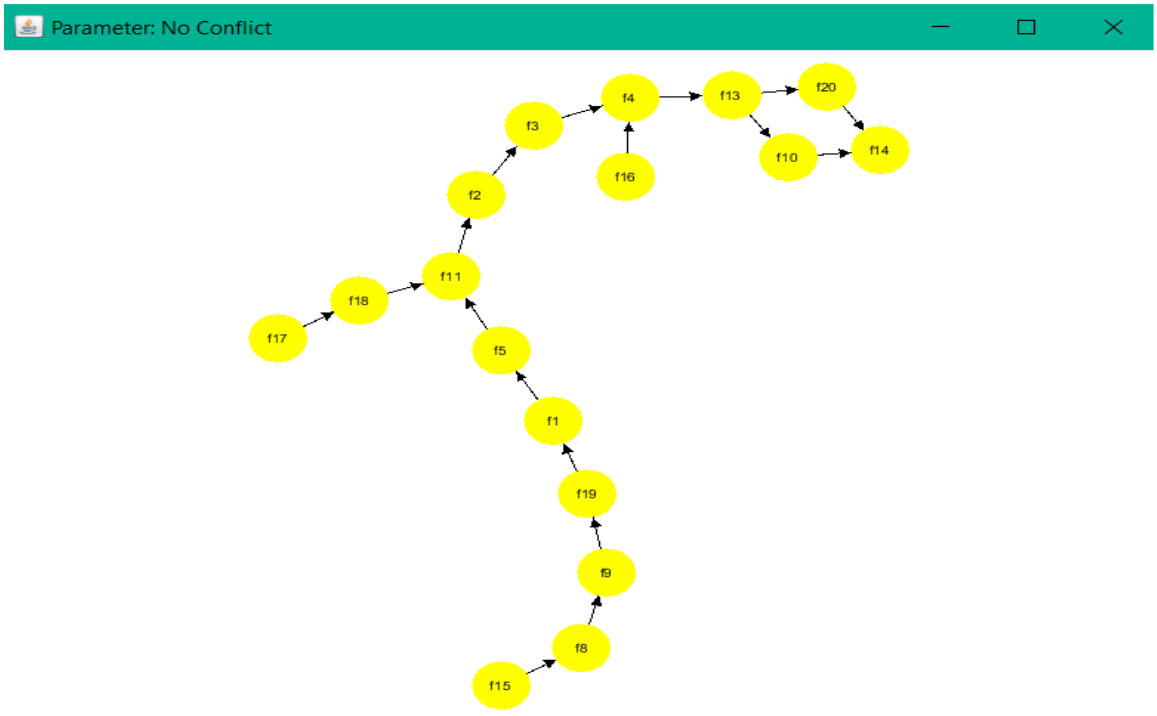

Figure 18: Partial order when conflict information is neglected

Summary:

In this document we present the partial order graphs generated for the different executions of each project.

The partial orders for all four parametric cases of our proposed approach are the same in this scenario (Figures 14, 15, 16 and 17). Although in this case we have two different macro clusters (for the two different NFR conflicts. Refer to requirements set), they share almost the same set of FRs. In addition to this, almost every pair of FR shares some common sets of NFR. Thus the partial order graphs are the same in all the cases.

A partial order without considering the NFR conflict information is also generated (Figure 18). It can be observed that ordering among the FRs changes when NFR conflicts are ignored. But some of the precedence relations are preserved (FRs f5 and f8 have the same precedence in all scenarios as they share a single NFR) due to NFR priority information. Now let us consider the FRs f3 and f4. In the first four parametric cases order is f4 precedes f3 (due to more than one conflict between their respective NFRs. Refer to requirements set). When conflict is ignored the order changes to f3 precedes f4 considering only NFR priority and FR-NFR dependency values.

In this document we present the partial order graphs generated for the different executions of each project.

DATA SET: PROMISE **PROJECT ID:** 2

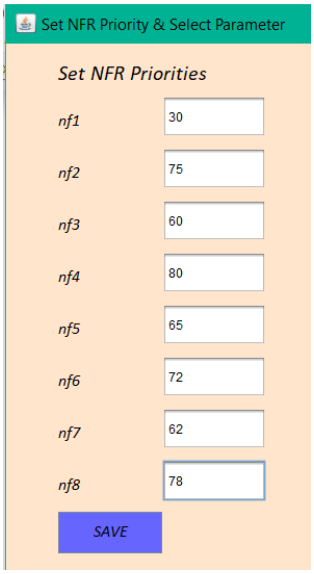

Figure 19: NFR priority values assigned

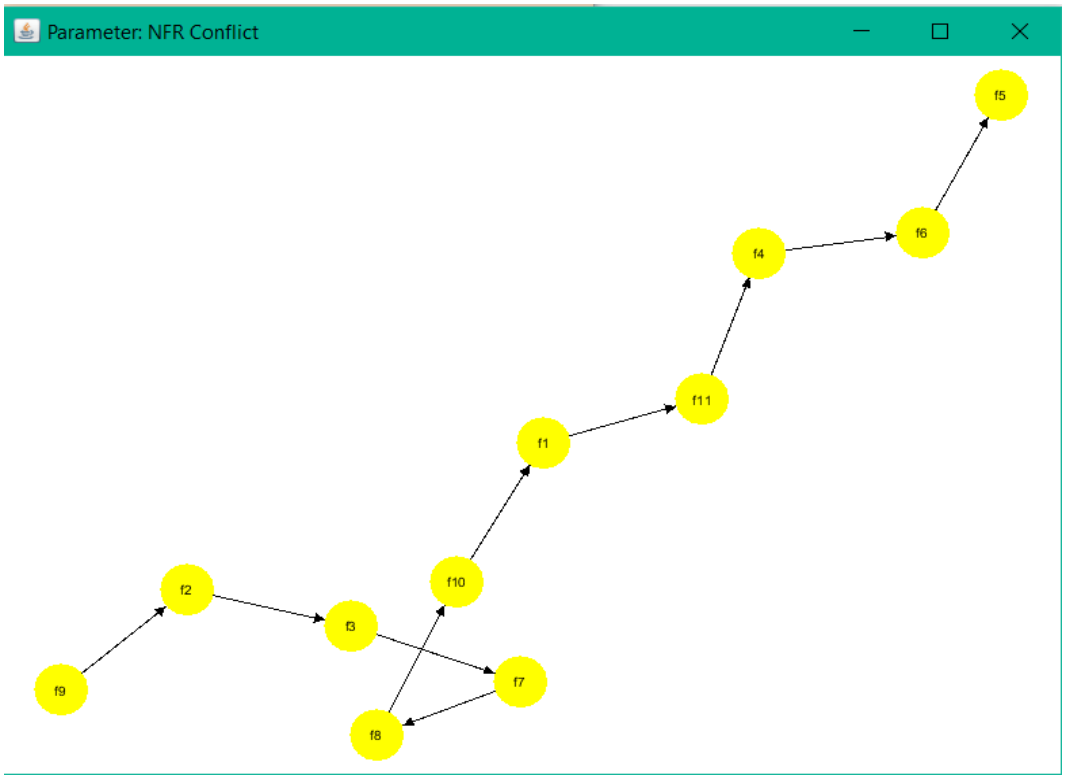

Figure 20: Partial order when parameter is NFR conflict

In this document we present the partial order graphs generated for the different executions of each project.

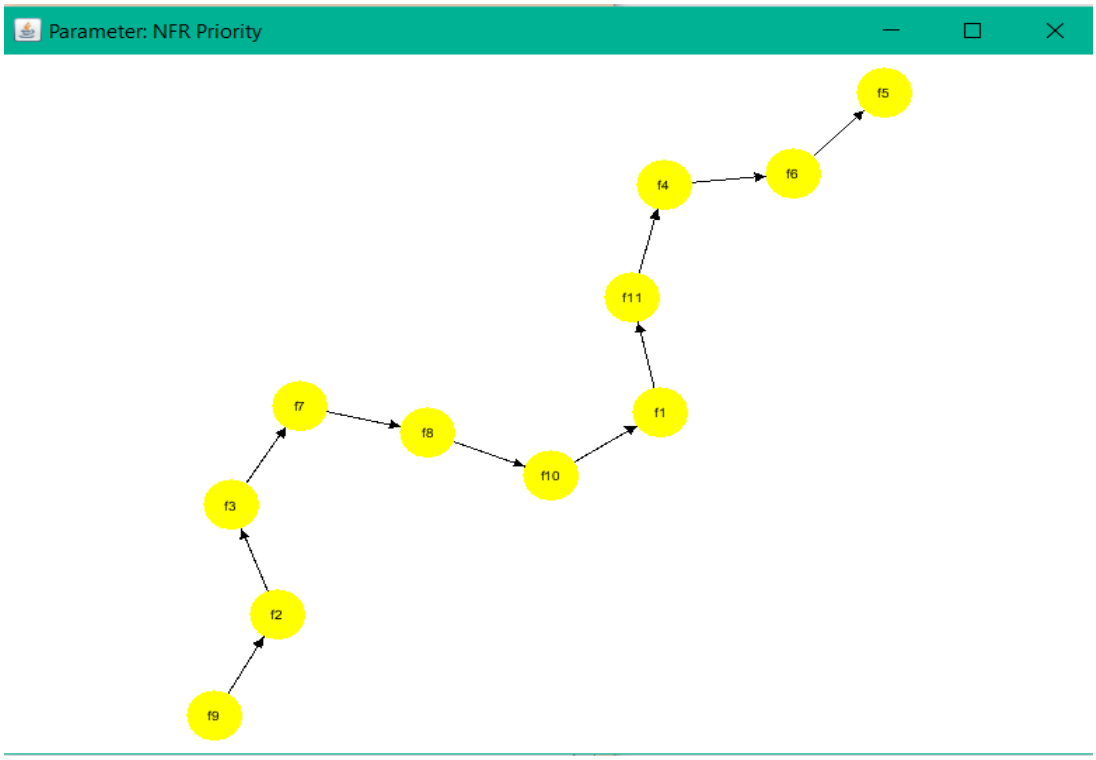

Figure 21: Partial order when parameter is NFR priority

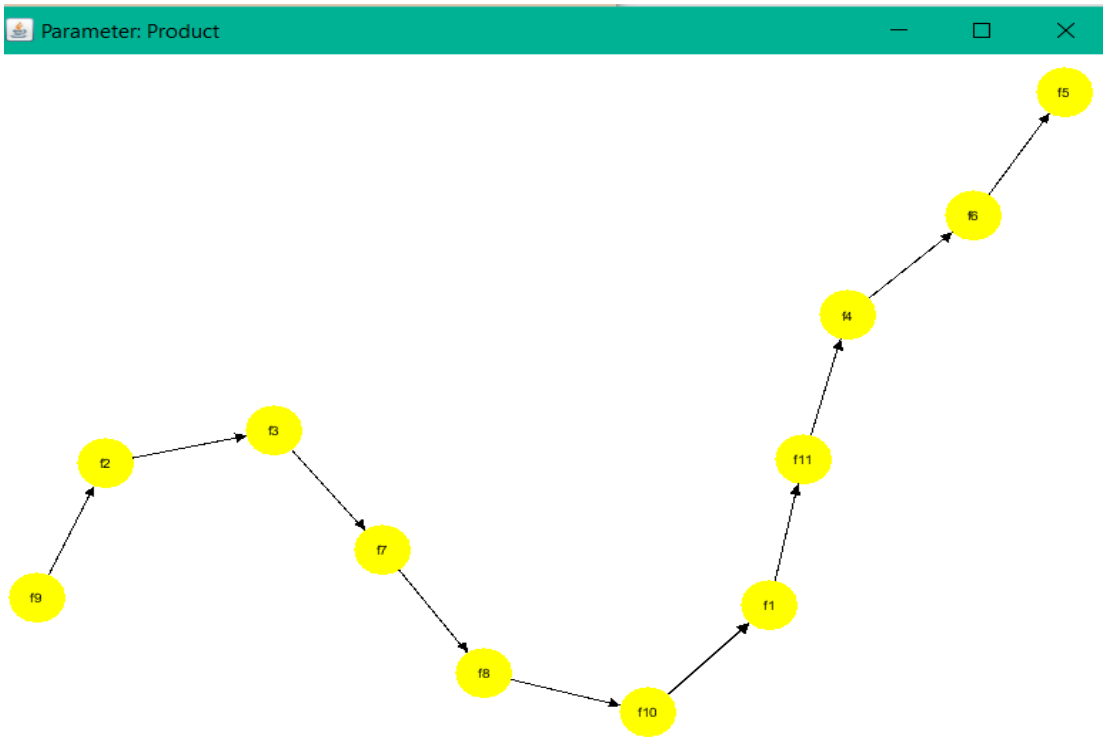

Figure 22: Partial order when parameter is product of NFR priority and conflict

In this document we present the partial order graphs generated for the different executions of each project.

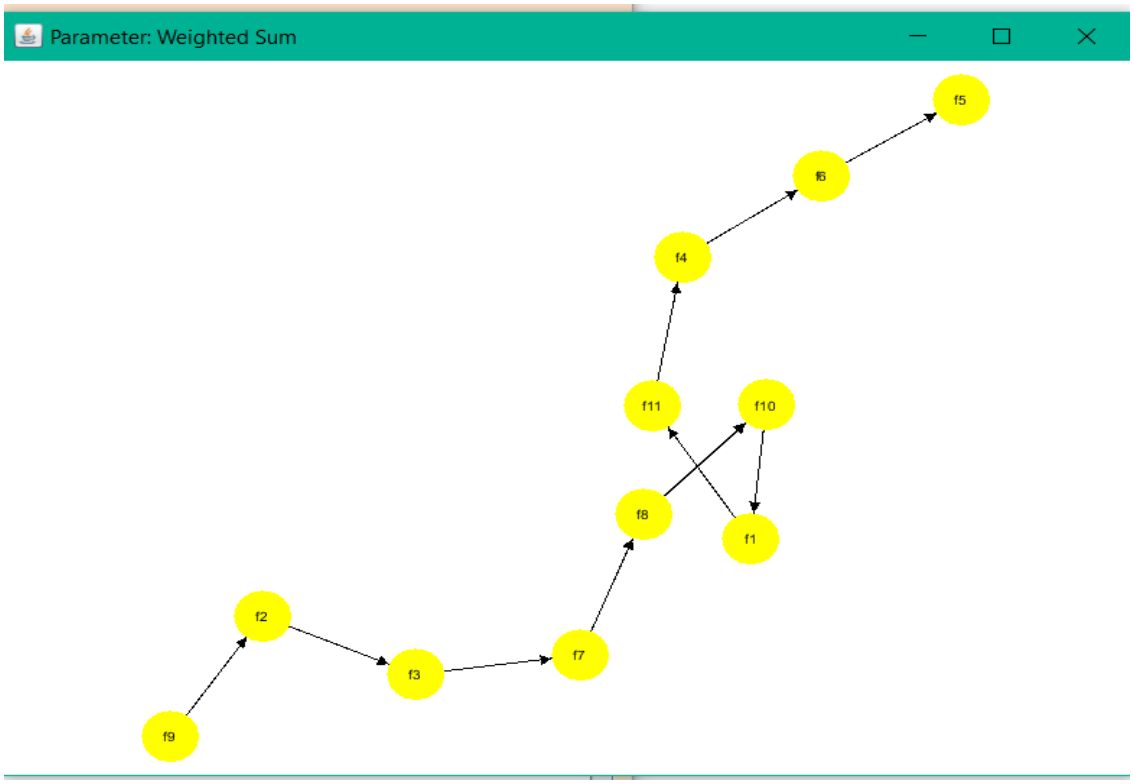

Figure 23: Partial order when parameter is weighted sum of NFR priority and conflict

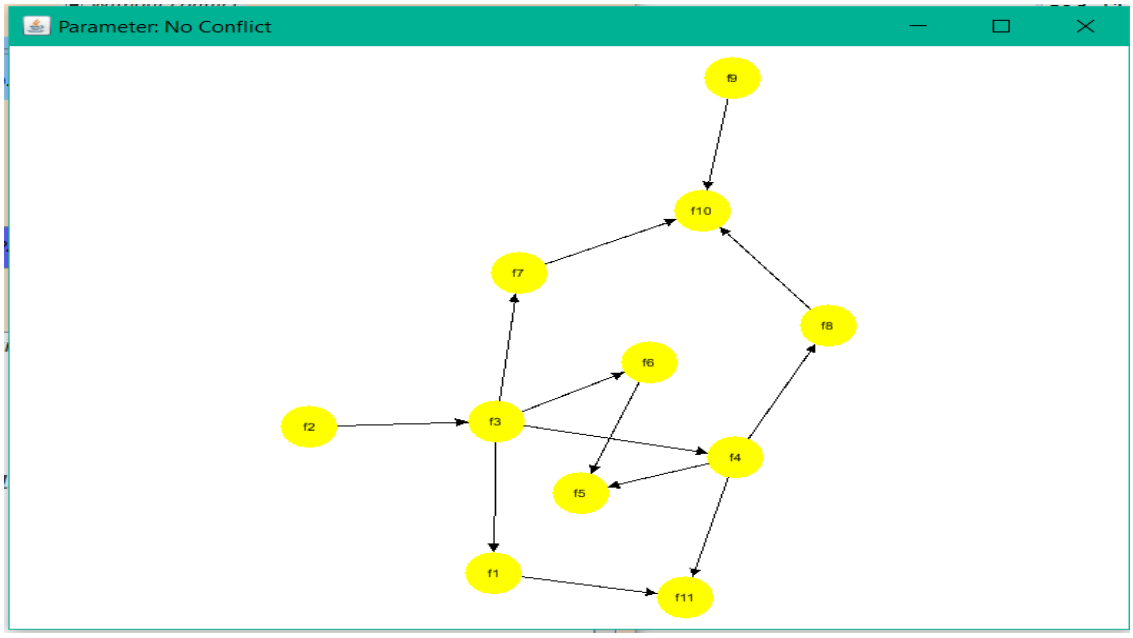

Figure 24: Partial order when conflict information is neglected

Summary:

In this document we present the partial order graphs generated for the different executions of each project.

The partial orders for all four parametric cases of our proposed approach are the same in this scenario (Figures 20, 21, 22 and 23). Although in this case we have four different macro clusters (for the four different NFR conflict. Refer to requirements set) they share the common same set of FRs.

A partial order without considering the NFR conflict information is also generated (Figure 24). It is clearly evident from the figure that ordering among FRs changes **significantly**. Let us consider the functionalities f4 and f11. In all the four parametric cases f11 precedes f4 due to conflict among their respective NFRs (performance & portability; performance & security). When conflict information is neglected the order changes to f4 precedes f11. This is because the order is determined considering only the NFR priority and FR-NFR dependency values.

In this document we present the partial order graphs generated for the different executions of each project.

DATA SET: PROMISE **PROJECT ID:** 5

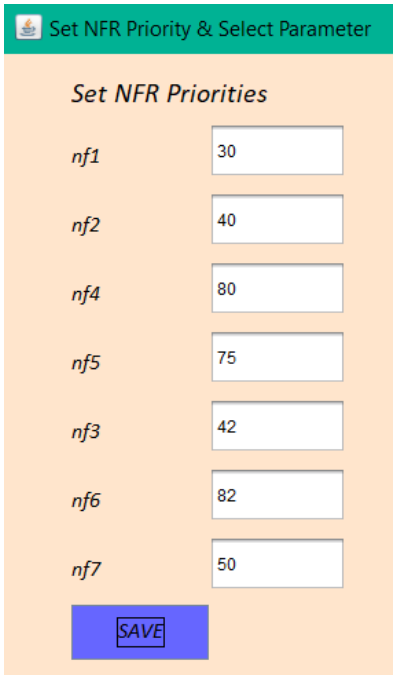

Figure 25: NFR Priority values assigned

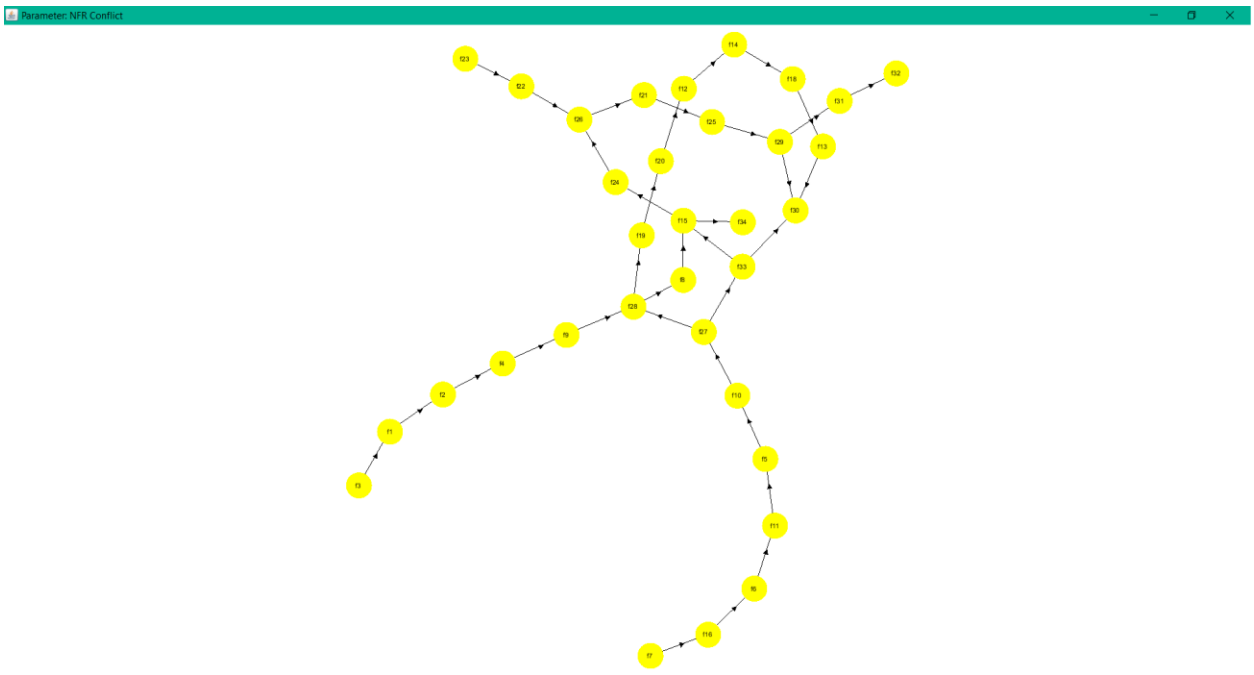

Figure 26: Partial order when parameter is NFR conflict

In this document we present the partial order graphs generated for the different executions of each project.

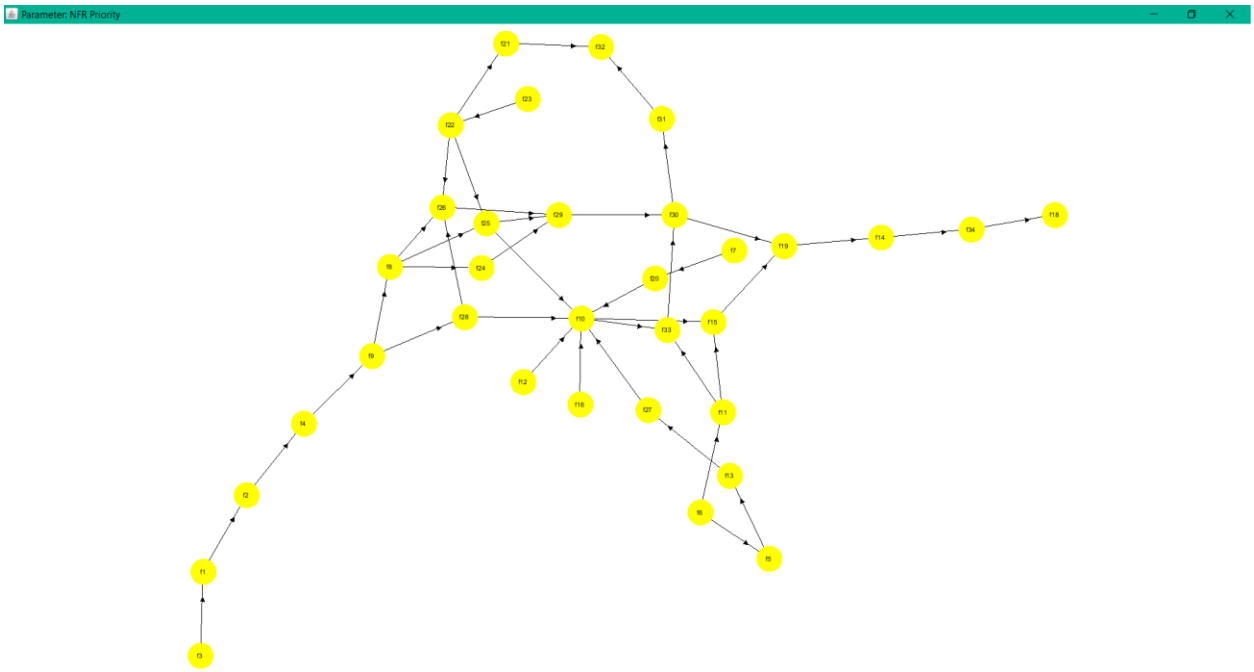

Figure 27: Partial order when parameter is NFR Priority

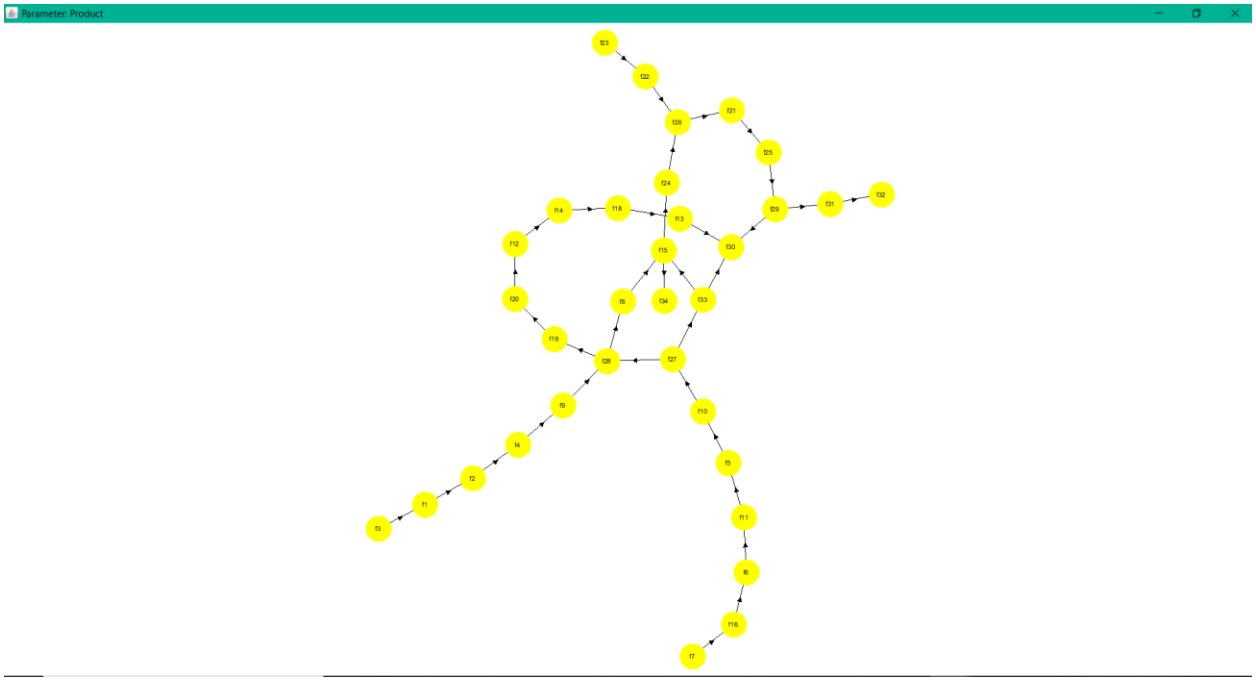

Figure 28: Partial order when parameter is product of NFR priority and conflict

In this document we present the partial order graphs generated for the different executions of each project.

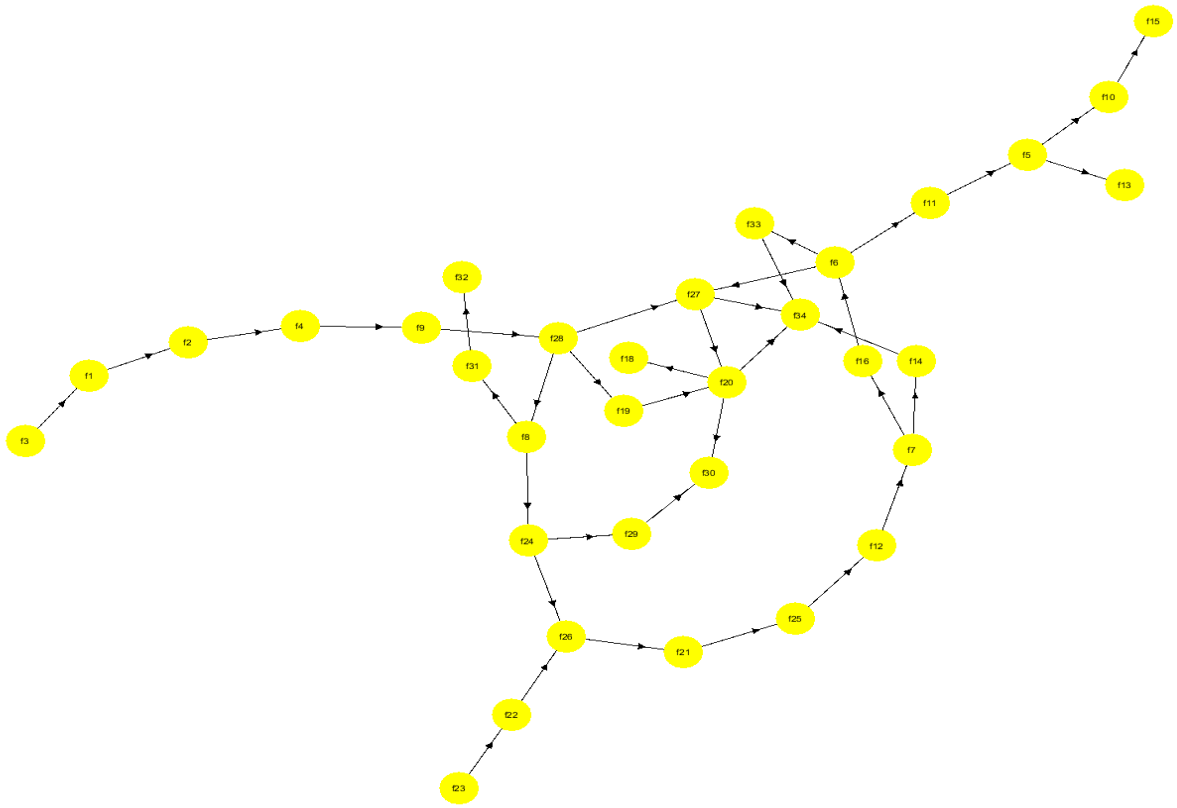

Figure 29: Partial order when parameter is weighted sum of NFR priority and conflict

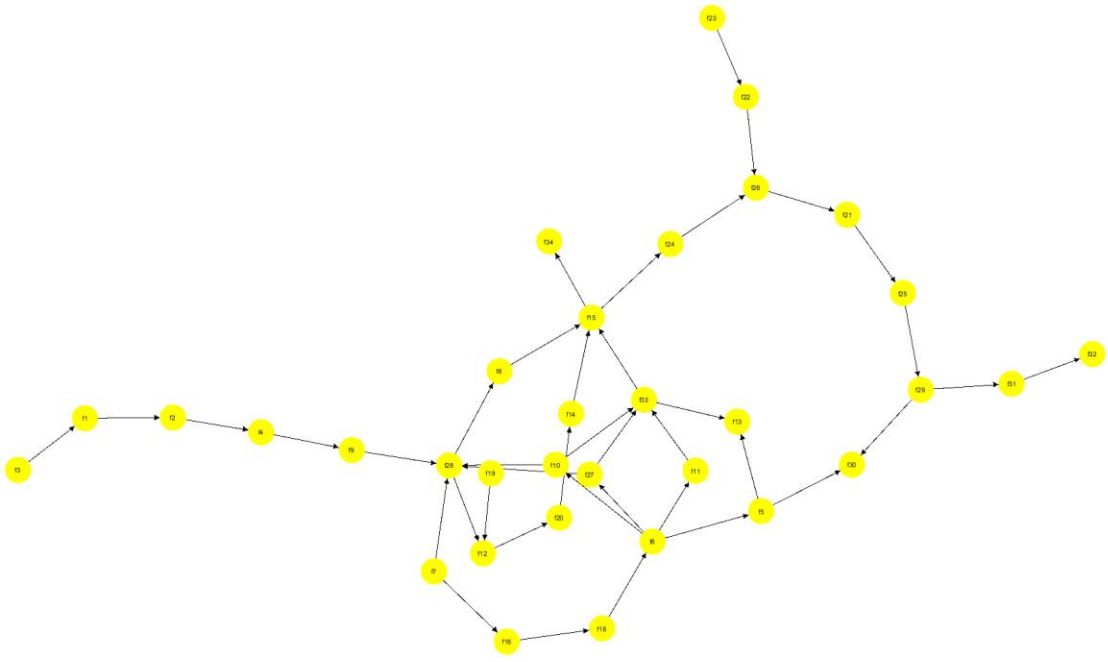

Figure 30: Partial order when conflict information is neglected

In this document we present the partial order graphs generated for the different executions of each project.

Summary:

In this case partial order graphs are identical when the parameter is either NFR conflict or product (Figures 26 and 28). The partial order graph generated when the parameter is NFR priority and weighted sum has some of its precedence relation different.

In figure 30 we show the partial order graph generated when NFR conflict information is completely neglected. It can be observed that precedence relation changes significantly.

N.B: While generating partial order for all the projects using *weighted sum* as a parameter we have used conflict weight as 0.2 and NFR-priority weight as 0.8.

In this document we present the partial order graphs generated for the different executions of each project.

The following table summarizes the number of partial order edges in each case and time and space consumed.

Name of the Project	Parameter	No. of partial order edges	Time(ms)	Memory Space (B)
PURE-Video Search	Conflict	6	10	10250848
PURE-Video Search	NFR Priority	6	3	23210752
PURE-Video Search	Product	6	4	24192688
PURE-Video Search	Weighted Sum	6	3	17256008
PURE-Video Search	No Conflict Only Priority	6	37	11223864
PURE-Digital Home	Conflict	10	24	10906352
PURE-Digital Home	NFR Priority	10	12	11608248
PURE-Digital Home	Product	10	12	12078656
PURE-Digital Home	Weighted Sum	10	23	11559536
PURE-Digital Home	No Conflict Only Priority	10	13	10877480
PROMISE-PID: 1	Conflict	17	157	12557288
PROMISE-PID: 1	NFR Priority	17	47	11765536
PROMISE-PID: 1	Product	17	85	10644664
PROMISE-PID: 1	Weighted Sum	17	42	2874872
PROMISE-PID: 1	No Conflict Only Priority	17	50	9702464
PROMISE-PID: 2	Conflict	10	109	12648296
PROMISE-PID: 2	NFR Priority	10	39	15323344
PROMISE-PID: 2	Product	10	39	12570184
PROMISE-PID: 2	Weighted Sum	10	82	28024160
PROMISE-PID: 2	No Conflict Only Priority	14	15	13064632
PROMISE-PID: 5	Conflict	35	313	16448704
PROMISE-PID: 5	NFR Priority	42	290	16308890
PROMISE-PID: 5	Product	35	350	16725808
PROMISE-PID: 5	Weighted Sum	38	376	17780088
PROMISE-PID: 5	No Conflict Only Priority	40	262	16677168