

Rapport de Projet d'Aide à la Décision

Programme de résolution de problème par les techniques de PSC



Groupe D3

Valentin Bauer
Timothée Marguier
Charles Frappa
Océane Fradon
Clémence Issanes

Pour ce projet nous devons réaliser un programme en C++ permettant la résolution d'un problème avec les techniques de PSC vues en cours, donc décrit au moyen de variables, domaines de valeurs et contraintes.

Le programme que nous avons réalisé permet la lecture d'un fichier contenant une liste des variables, les domaines de valeurs associés et les contraintes, et la résolution du problème associé avec 4 les méthodes qui sont demandées.

Note importante au correcteur : Il se peut qu'un problème de compilateur survienne, n'hésitez pas alors à utiliser le compilateur C++11.

1. Structure de notre code
2. Résultats obtenus
3. Conclusion

1. Structure de notre code

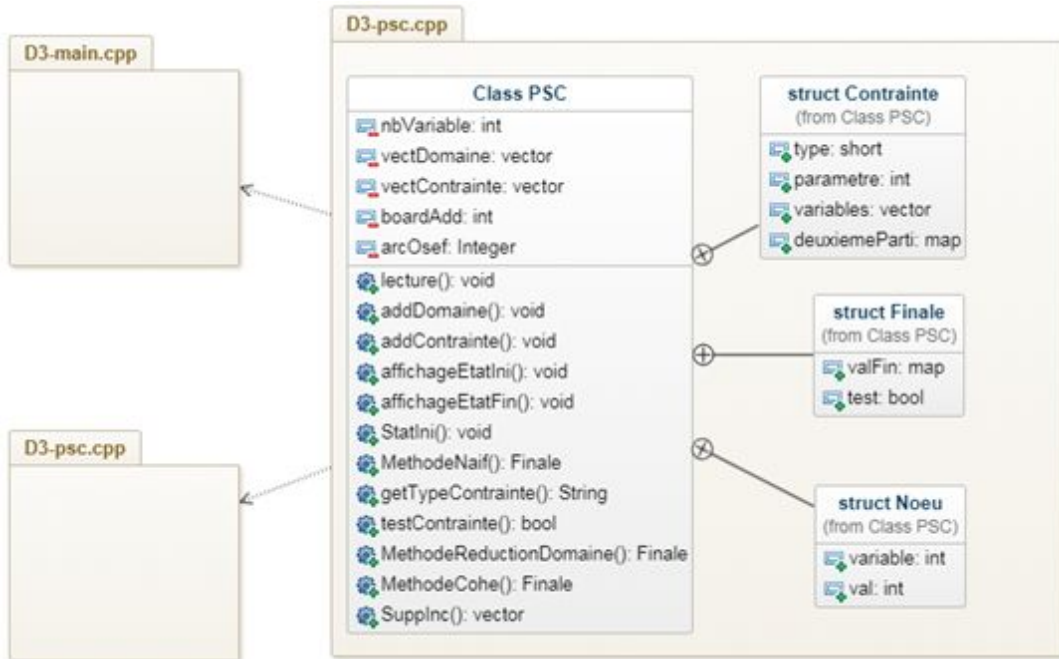
Afin de résoudre ce problème, notre projet va travailler de la façon suivante :

- Dans un premier temps, nous demandons à l'utilisateur de rentrer le nom du fichier que nous souhaitons analyser
- Nous allons ensuite récupérer les contraintes et les variables de ce fichier avec la fonction lecture puis les ajouter à notre programme.
- Une fois récupérées, nous allons afficher nos différentes variables avec la fonction affichageEtatIni ainsi que nos domaines et nos différentes contraintes.
- Nous allons ensuite demander à l'utilisateur de sélectionner la méthode de résolution qu'il veut appliquer et en fonction de ce choix nous appelons les fonctions

correspondantes (*methodenaif*, *methodereductiondomaine*, *methodeoptimisation*, *methodecohe*).

- Chacune de ces fonctions va tenter de résoudre le problème avec la méthode du choix de l'utilisateur.
- Enfin, si la fonction a trouvé un résultat en fonction des contraintes données, alors nous allons afficher la valeur finale de chaque variable ainsi que le nombre de noeuds créés et le nombre de branches ignorées.

Architecture du programme :



Pour ce qui est des contraintes reconnues par notre programme, elles sont les suivantes :

```

string PSC::getTypeContrainte(int type) {
    switch (type) {
        case 1:
            return "=";
        case 2:
            return "!=";
        case 3:
            return "<=";
        case 4:
            return "Somme (Xi) =";
        case 5:
            return "<";
        case 6:
            return ">";
        case 7:
            return "Somme (Xi) <";
        case 8:
            return ">=";
        case 9:
            return "Somme (Xi) = Ai * Bi";
        default:
            return "";
    }
}

```

Elles ont été créées en fonction des problèmes qui nous ont été posés.

2. Résultats obtenus

Afin de condenser le temps de réponse et les résultats obtenus avec les différents exercices, nous avons regroupés ces résultats dans un tableau que voici.

Il permet donc de regrouper les résultats des différents exercices (de 1 à 8) et leurs résultats en fonction de la méthode de résolution triviale, de la méthode de résolution des domaines de valeurs, de la méthode de résolution par la variable la plus contrainte et de la méthode de cohérence d'arête.

exo		1	2	3	4	5	6	7	8
triv	nœuds créés	long	long	long	long	long	long	long	long
	branches ignorées	long	long	long	long	long	long	long	long
reduc	nœuds créés	539520	long	long	67	7	long	long	16
	branches ignorées	1071776	long	long	82	6	long	long	155
variable	nœuds créés	3692	long	long	37	9	long	long	16
	branches ignorées	143387	long	long	36	6	long	long	155
arete	nœuds créés	510	510	long	29	8	84	1	16
	branches ignorées	0	0	long	0	0	0	0	0
					pas de res	pas de res		pas de res	

3. Conclusion

Dans de nombreux cas, certaines méthodes sont trop longues et nos machines ne nous permettent pas toujours de venir au bout du problème. Cela est sûrement dû à une mauvaise optimisation de notre projet. Cependant, nous avons réussi à réaliser les 4 méthodes malgré un temps d'exécution particulièrement long notamment pour la méthode triviale qui est la plus longue de toutes.

