

Métaheuristique

—

Sport Tournament Scheduling

Cyril Grelier

Métaheuristique
Master 2 ID
Université d'Angers
Faculté des sciences
Département informatique

Février 2020



Sommaire

- 1 Introduction
 - STS
 - État de l'art
- 2 Implémentation
- 3 Méthodes
 - Recherche locale
 - Algorithmes génétiques
- 4 Tests
- 5 Conclusion

Problème étudié

Sport Tournament Scheduling (STS)

- Problème 26 du CSPLib¹
- Problème de rencontres sportives :
 - ▶ n équipes ($n \bmod 2 = 0$), s semaines ($s = n - 1$), p périodes ($p = n/2$)
 - ▶ Chaque équipe rencontre toutes les autres
 - ▶ Chaque équipe doit jouer une fois par semaine
 - ▶ Chaque équipe ne joue pas plus de deux fois par période

	Période 0	Période 1	Période 2	Période 3
Semaine 0	0 vs 1	2 vs 3	4 vs 5	6 vs 7
Semaine 1	0 vs 2	1 vs 7	3 vs 5	4 vs 6
Semaine 2	4 vs 7	0 vs 3	1 vs 6	2 vs 5
Semaine 3	3 vs 6	5 vs 7	0 vs 4	1 vs 2
Semaine 4	3 vs 7	1 vs 4	2 vs 6	0 vs 5
Semaine 5	1 vs 5	0 vs 6	2 vs 7	3 vs 4
Semaine 6	2 vs 4	5 vs 6	0 vs 7	1 vs 3

1. CSPLib Problem 026 : Sports Tournament Scheduling, Toby Walsh,
<http://www.csplib.org/Problems/prob026>

État de l'art

Différentes approches

- programmation linéaire en nombres entiers (jusqu'à 12 équipes),
- différentes versions de recherche locale (jusqu'à 20 équipes),
- programmation par contraintes (jusqu'à 30 équipes),
- combinaison de recherche locale avec tabou et programmation par contrainte (jusqu'à 40 équipes),
- algorithme de réparation (jusqu'à 40 équipes),
- algorithme énumératif (jusqu'à 70 équipes).

Implémentation

Représentation

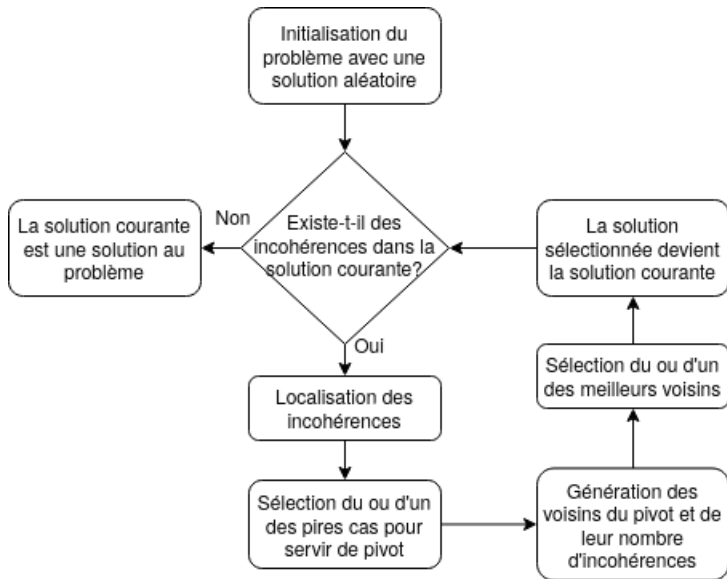
- Planning
- Matrices d'incohérences

$[(1, 3), (0, 3), (1, 2)],$	$[[2, 1, 1],$	$[[3, 1, 1],$
$[(3, 4), (3, 5), (0, 5)],$	$[1, 2, 1],$	$[2, 3, 2],$
$[(2, 3), (4, 5), (0, 2)],$	$[1, 0, 1],$	$[2, 1, 2],$
$[(2, 4), (1, 4), (0, 4)],$	$[1, 1, 1],$	$[1, 1, 2],$
$[(0, 1), (2, 5), (1, 5)]]$	$[1, 1, 2]]$	$[1, 2, 2]]$

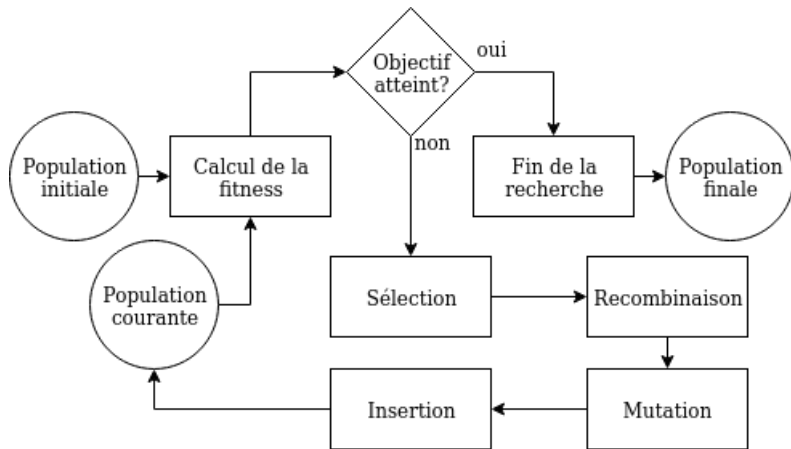
Fonctions

- Génération de matrices d'incohérences (semaines, périodes, totale)
- Calcul du nombre d'incohérences (semaines, périodes, totale)
- Génération du voisinage à partir d'un point (totale, colonne)

Recherche locale et recherche locale avec tabou



Algorithmes génétiques et mémétiques



Tests

	Recherche locale		Recherche locale avec tabou		Recherche locale modifiée		Algorithmes génétiques		Algorithmes mémétiques		Algorithmes mémétiques modifiés	
6	97	>1s	95	>1s	100	>1s	75	3min	100	6s	100	3s
8	89	7s	100	2s	97	2s	18	9min	100	40s	100	11s
10	28	68s	86	35s	98	22s	0	15min	100	4min	100	159s
12	3	4min	21	183s	24	86s	-	-	100	2h	100	1h30
14	0	16min	0	6min	1	193s	-	-	2/2	1h15	5/10	1h45
16	-	-	-	-	1	6min	-	-	0/2	15h	0/10	18h45

Conclusion

Améliorations possibles

- Forcer certains matches
- Optimisation du code des algorithmes génétiques
- Rapprocher le fonctionnement des algorithmes génétiques au problème :
 - ▶ Lors de la mutation, augmenter les chances de muter les mauvais matches
 - ▶ Éviter le crossover sur des semaines valides
 - ▶ Trouver un moyen de garder une diversité correcte sans insérer de trop mauvais individus
 - ▶ Trouver d'autres crossover/mutation efficaces
- Permutations sur les périodes en partant d'une configuration valide sur les semaines