

Nos primeiros testes que realizamos, e que foram entregues com o trabalho, estávamos trabalhando com uma mutação que fazia uma recombinação de jogos times para uma determinada rodada com probabilidade m . Com estes testes, havíamos obtido os seguintes resultados:

Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
-	NL4	12	0.3	0.1	0.3	3	5	0.002785	4	8559	8276	96.69%
-	super4	12	0.3	0.1	0.3	3	5	0.001533	4	89317	63405	70.99%
-	super4	12	0.3	0.1	0.3	15	5	0.005872	16	89317	63405	70.99%
-	super4	12	0.3	0.1	0.3	50	5	0.022853	51	89317	63405	70.99%
-	super4	12	0.3	0.1	0.3	1000	5	0.285921	1001	89317	63405	70.99%
-	super4	12	0.3	0.1	0.3	2000	5	0.561663	2001	89317	63405	70.99%
-	super4	30	0.3	0.1	0.3	1000	5	0.426776	1001	64270	63405	98.65%
-	super4	35	0.3	0.1	0.3	2000	5	1.14921	2020	63405	63405	100.00%
-	galaxy4	30	0.3	0.1	0.3	1000	5	0.801635	1786	437	416	95.19%
-	galaxy4	30	0.3	0.1	0.3	2000	5	1.2361	2786	437	416	95.19%
-	galaxy4	60	0.3	0.1	0.3	1000	5	0.839437	1001	421	416	98.81%
-	galaxy4	60	0.3	0.1	0.3	2000	5	1.69821	2001	421	416	98.81%

- t - Quantidade de times,
- M - Matriz que representa uma instância do problema,
- T - Tamanho da população de entrada,
- c - Taxa de filhos gerados a partir do crossover na próxima geração,
- p - Taxa de filhos gerados a partir da operação de mutação na próxima geração,
- m - Taxa de mutação que ocorrerá dentro do cromossomo,
- d - Quantidade máxima aceitável de gerações sem melhora (para o critério de parada),
- s - Tempo máximo de execução (para o critério de parada).

Exemplo de chamada: ./traboc_ttpga -o NL4_1.sol -t 4 -T 100 -p 0.6 -c 0.1 -m 0.30 -d 10000 -s 5 -M ATL,NYM,PHI,MON/0,745,665,929/745,0,80,337/665,80,0,380/929,337,380,0

Por causa do problema que nós tivemos com o memory leak do nosso programa, não conseguimos realizar nenhum teste para instâncias maiores. Para realizar a correção, utilizamos o *valgrind*, que nos deu um relatório mais detalhado dos problemas de memória que estavam ocorrendo no programa.

Com o programa corrigido para instâncias maiores, optamos por alterar também a operação de mutação. Ao invés de realizarmos uma permutação com todos os times dentro de uma determinada rodada, faremos a troca entre dois jogos distintos em rodadas aleatórias. Este processo também ocorre com uma probabilidade m . Com esta nova operação de mutação, obtivemos os seguintes resultados para as mesmas instâncias já testadas anteriormente:

Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
NL4_1	NL4	100	0.6	0.3	0.1	10000	5	12.6425	10464	8276	8276	100.00%
NL4_2	NL4	100	0.6	0.3	0.1	1000	5	1.75442	1464	8276	8276	100.00%
NL4_3	NL4	100	0.6	0.3	0.1	500	5	1.15595	964	8276	8276	100.00%
NL4_4	NL4	50	0.6	0.3	0.1	500	5	0.335815	523	8559	8276	96.69%
Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
Super4_1	super4	50	0.6	0.3	0.1	500	5	0.423932	645	63405	63405	100.00%
Super4_2	super4	50	0.6	0.3	0.1	300	5	0.290369	445	63405	63405	100.00%
Super4_3	super4	30	0.6	0.3	0.1	100	5	0.065863	189	88594	63405	71.57%
Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
galaxy4_1	galaxy4	100	0.6	0.3	0.1	10000	5	12.2492	10042	416	416	100.00%
galaxy4_2	galaxy4	100	0.6	0.3	0.1	500	5	0.659651	542	416	416	100.00%
galaxy4_3	galaxy4	50	0.6	0.3	0.1	500	5	0.335337	523	468	416	88.89%

Conforme vemos na tabela, apesar de trabalharmos com populações iniciais maiores, obtivemos uma melhora significativa no resultado, sendo possível chegar no resultado ótimo muito mais rápido, ou seja, com uma quantidade de gerações muito menor.

O próximo teste que nós realizamos foi das instâncias de tamanho 6, onde foi possível observar que a nossa operação de mutação ainda estava ineficaz:

Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
NL6_1	NL6	100	0.6	0.3	0.1	100000	5	300.001	79817	10^{11}	22969	0.00%
NL6_2	NL6	500	0.7	0.3	0.1	1000000	10	600.013	33487	10^{11}	22969	0.00%

Para tentar solucionar este problema, discutimos quais alterações que nós poderíamos fazer para lidar com a restrição da classe 10^{11} , que representa dois jogos AxB e BxA ocorrendo em rodadas consecutivas. Realizamos então mais uma alteração na nossa operação de mutação para que, quando ele fosse realizar a troca entre dois jogos AxB e CxD, verificasse se na rodada anterior ao jogo AxB não existe uma partida entre Dx C. Caso exista, tenta selecionar outra rodada para trocar. Com esta nova operação de mutação, tivemos os seguintes resultados:

Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
NL6_1_cc2	NL6	500	0.7	0.4	0.1	100000	10	600.001	32802	27036	22969	84.96%
NL6_2_cc2	NL6	150	0.7	0.4	0.1	100000	10	575.488	111342	10^{11}	22969	0.00%
NL6_3_cc2	NL6	500	0.7	0.4	0.1	1000000	20	1200.02	42976	27036	22969	84.96%
NL6_4_cc2	NL6	500	0.7	0.3	0.1	1000000	20	1200.01	43888	10^{11}	22969	0.00%
NL6_5_cc2	NL6	500	0.7	0.2	0.1	1000000	20	1200.02	43040	28007	22969	82.01%
NL6_6_cc2	NL6	500	0.7	0.1	0.1	1000000	20	1200.01	43677	28149	22969	81.60%
NL6_7_cc2	NL6	500	0.7	0.4	0.1	1000000	30	1800.01	93104	26228	22969	87.57%
NL6_8_cc2	NL6	500	0.7	0.7	0.1	1000000	30	1800	92848	27452	22969	83.67%
NL6_9_cc2	NL6	500	0.7	0.4	0.1	1000000	60	3600.01	169994	26228	22969	87.57%
NL6_10_cc2	NL6	500	0.7	0.4	0.1	1000000	120	7200.02	335382	26228	22969	87.57%
NL6_11_cc2	NL6	500	0.7	0.6	0.1	1000000	120	7200	293132	27791	22969	82.65%
Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
Galaxy6_1	galaxy6	500	0.7	0.4	0.1	1000000	30	1800.01	72035	1572	1365	86.83%
Galaxy6_2	galaxy6	250	0.7	0.5	0.1	1000000	60	3600.01	149162	1514	1365	90.16%
Galaxy6_3	galaxy6	500	0.7	0.2	0.1	1000000	60	3600	308568	10^{11}	1365	0.00%
Galaxy6_4	galaxy6	500	0.7	0.2	0.1	1000000	120	7200	366593	1447	1365	94.33%
Nome Tst	Instância	-T	-p	-m	-c	-d	-s	Tempo	Total Ger.	Fitness	Sol. Ótima	% Otim.
super6_1	super6	500	0.7	0.4	0.1	1000000	30	1800.01	71952	161466	130365	80.74%
super6_2	super6	500	0.7	0.4	0.1	1000000	60	3600	158091	160210	130365	81.37%
super6_3	super6	500	0.7	0.6	0.1	1000000	60	3600	190671	10^{11}	130365	0.00%
super6_4	super6	500	0.7	0.1	0.1	1000000	120	7200	294621	148180	130365	87.98%

No teste NL6_8_cc2 e super6_3 podemos notar que a nossa solução piora com o aumento brusco de -m que é a taxa de mutação dentro do cromossomo. Nas três instâncias testadas, observamos uma melhora significativa sempre que aumentamos os critérios de parada, ou seja, sempre que o algoritmo alcançou gerações maiores. Realizamos alguns testes com populações iniciais maiores, mas não obtivemos melhoras na *fitness*, principalmente pelo fato de que é necessário muito mais tempo para percorrer as gerações.