6
Experimentos numéricos

Os experimentos numéricos foram realizados em MATLAB 6 e executados em um computador com processador de 2,4 GHz e 2 GB de memória RAM.

6.1
Experimentos numéricos para o FSP

Para resolver o FSP usando o método proposto, foram aplicados os algoritmos das seções 5.1.1 – 5.1.2, com as funções objetivo $f_1 = c_M$ =*Makespan*, $f_2 = \bar{F}$ = Tempo de fluxo total.

As instâncias usadas nos testes foram tomadas de Taillard (1993). Cada instância é representada por $K \times N$, onde K é o número de tarefas e N o número de máquinas. Neste estudo, serão testadas as instâncias TA31, TA41, TA51, TA61 e TA71. Os resultados obtidos pelo método proposto nestas instâncias serão comparados com os melhores resultados dos algoritmos: PG-ALS (PASUPATHY *ET AL.*, 2006), MOGLS (MURATA *ET AL.*, 1996), ENGA (BAGCHI, 1999) e GPWGA (CHANG *ET AL.*, 2002). O algoritmo proposto foi aplicado considerando 100 soluções iniciais com 10 execuções para cada instância. Não foi possível a comparação de tempos, já que não foram reportados os tempos de execução nos algoritmos comparados. Adicionalmente, para a instância TA51: 50×20 o algoritmo proposto considerou testes com 200 e 400 soluções iniciais para analisar a convergência.

A seguir, são apresentados os resultados obtidos com o método proposto e pelos métodos existentes acima mencionados. Cada solução *não-dominada* é representada pelo par $(c_M, \bar{F})$. Logo, na Tabela 8, as melhores soluções *não-dominadas* encontradas pelos algoritmos existentes são comparadas com as soluções *não-dominadas* encontradas pelo algoritmo proposto. A mesma lógica para apresentação dos resultados é usada para outras instâncias, em tabelas que serão apresentadas adiante.

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 8 – Resultados computacionais para a instância TA31: 50×5

Algoritmos Existentes		Método Proposto	
c_M	$\bar{F}$	c_M	$\bar{F}$
2.724	71.531	2.729	67.460
2.729	68.036	2.731	66.901
2.731	67.028	2.735	66.041
2.752	66.061	2.736	66.021
2.757	66.052	2.743	65.931
2.758	66.047	2.746	65.928
2.763	66.032	2.748	65.871
2.765	66.024	2.762	65.495
2.770	65.979	2.770	65.280
2.799	65.963	2.752	65.505

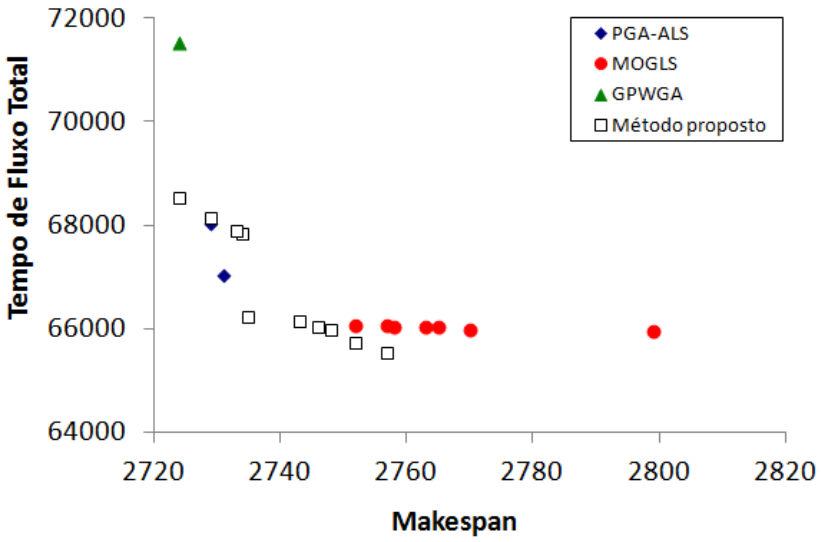

Figura 31 – Aproximação da *fronteira de Pareto* para a instância TA31: 50×5

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 9 – Resultados computacionais para a instância TA41: 50×10

Algoritmos Existentes				Método Proposto	
c_M	$\bar{F}$	c_M	$\bar{F}$	c_M	$\bar{F}$
3.047	93.511	3.133	90.663	3.072	92.115
3.052	93.013	3.134	90.641	3.080	91.797
3.059	92.666	3.135	90.448	3.084	91.241
3.063	92.602	3.137	90.408	3.098	91.023
3.070	92.508	3.148	90.364	3.099	90.981
3.074	92.493	3.152	90.305	3.106	90.955
3.075	92.124	3.156	90.254	3.120	90.656
3.076	91.757	3.197	90.207	3.141	90.628
3.087	91.688	3.209	90.165	3.142	90.557
3.097	91.256	3.237	90.158	3.146	90.520
3.099	91.236	3.249	90.099	3.147	90.428
3.111	91.149	3.298	90.075	3.154	89.538
3.132	90.882				

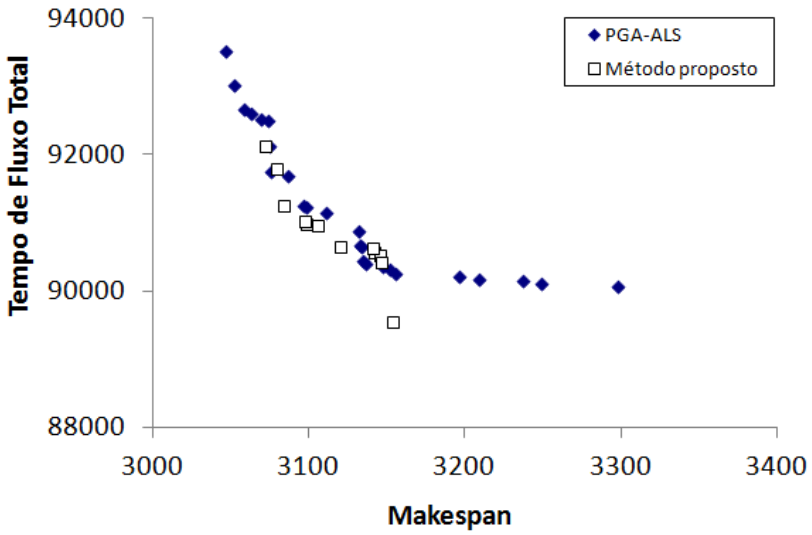

Figura 32 – Aproximação da *fronteira de Pareto* para a instância TA41: 50×10

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 10 – Resultados computacionais para a instância TA51: 50×20

Algoritmos Existentes		Método Proposto (100)		Método Proposto (200)		Método Proposto (400)	
c_M	$\bar{F}$	c_M	$\bar{F}$	c_M	$\bar{F}$	c_M	$\bar{F}$
3.962	132.782	3.972	137.371	3.956	134.898	3.955	133.467
3.965	133.658	3.973	136.114	3.969	133.596	3.978	133.317
3.965	132.133	3.975	135.923	3.974	133.556	3.981	132.881
3.967	132.131	3.978	134.930	3.978	133.317	3.985	132.867
3.969	131.922	3.981	132.881	3.981	132.881	3.991	132.466
3.971	131.728	3.985	132.867	3.985	132.867	3.992	131.963
3.973	131.538	3.991	132.466	3.991	132.466	4.006	130.560
3.984	131.387	4.031	131.969	4.006	130.560	4.014	130.411
3.996	131.378	4.049	131.580	4.014	130.411	4.035	129936
3.997	130.965	4.067	130.801	4.035	129.936	4.049	129.484
3.998	130.351			4.049	129.484	4.076	129.473
4.018	130.311					4.095	129.438
4.022	130.283						
4.030	130.076						
4.031	129.835						
4.036	129.807						
4.049	129.451						
4.068	129.436						
4.182	129.314						

PUC-Rio - Certificação Digital Nº 1012728/CA

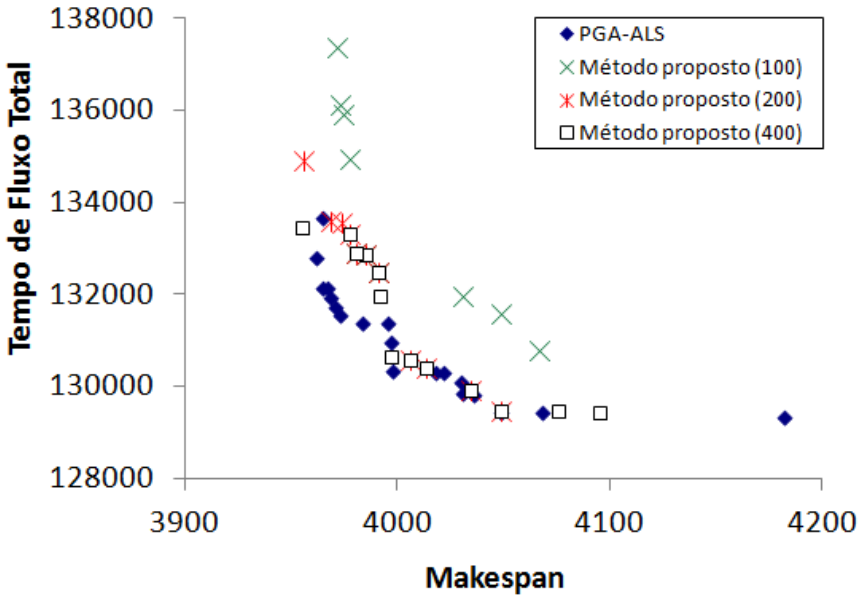

Figura 33 – Aproximação da *fronteira de Pareto* para a instância TA51: 50×20

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 11 – Resultados computacionais para a instância TA61: 100×5

Algoritmos Existentes		Método Proposto	
c_M	$\bar{F}$	c_M	$\bar{F}$
5.493	287.684	5.493	261.717
5.495	262.647	5.495	259.338
5.498	262.335	5.498	259.088
5.527	261.411	5.538	258.507
5.563	261.071	5.539	258.501
5.564	260.706		

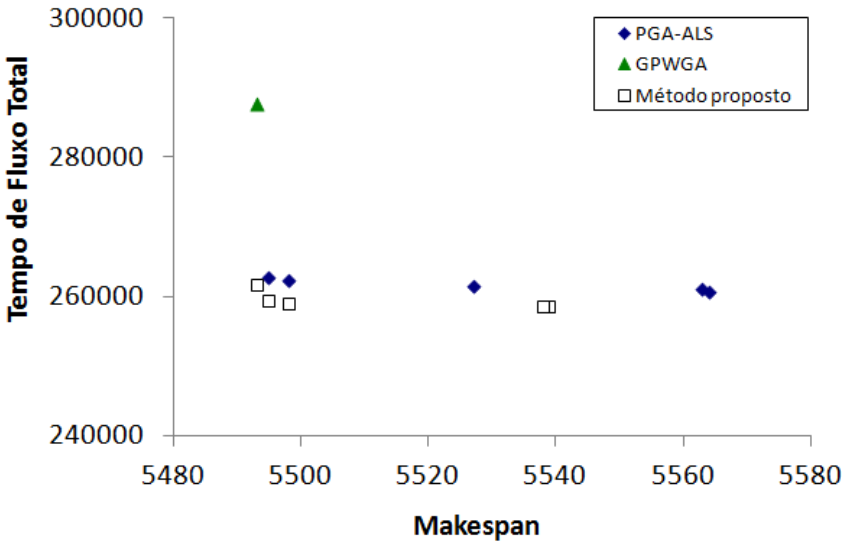

Figura 34 – Aproximação da *fronteira de Pareto* para a instância TA61: 100×5

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 12 – Resultados computacionais para a instância TA71: 100×10

Algoritmos Existentes				Método Proposto	
c_M	$\bar{F}$	c_M	$\bar{F}$	c_M	$\bar{F}$
5.801	325.462	5.858	314.749	5.836	318.588
5.803	324.725	5.877	312.785	5.842	313.791
5.804	318.924	5.881	312.632	5.843	313.790
5.806	318.299	5.892	312.534	5.848	313.769
5.816	318.055	5.897	312.349	5.849	313.208
5.827	316.972	5.904	312207	5.856	312.643
5.832	316.642	5.915	310.887	5.866	311.872
5.836	316.542	5.920	310.515	5.874	309.183
5.837	316.292	5.928	310.359	5.903	308.818
5.838	316.161	5.934	310.297	5.905	308.291
5.840	315.753	5.995	310.227	5.912	307.660
5.851	315.184	6.001	310.040	5.960	307.349
5.856	314.879	6.009	310.005		

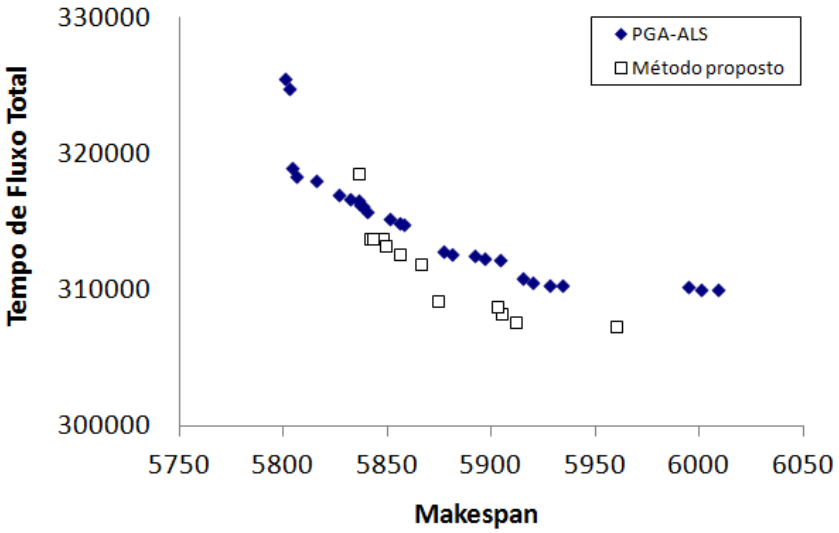

Figura 35 – Aproximação da *fronteira de Pareto* para a instância TA71: 100×10

Os resultados dos experimentos numéricos para o FSP mostram que o método proposto conseguiu boa aproximação da *fronteira de Pareto* e ainda obteve melhores resultados que os algoritmos existentes, mesmo para problemas de 50 e 100 tarefas, e número de máquinas inferior a 20. No entanto, não se conseguiu uma aproximação desejada na instância TA51 com 50 tarefas e 20 máquinas.

Os testes sugerem que, para instâncias com número de máquinas igual a 5 ou 10, 100 soluções inicias são suficientes para se obter bons resultados. Porém, quando o número de máquinas é superior a 10, os testes indicam que é necessário pelo menos 200 soluções inicias, o que aumenta significativamente o esforço computacional, para encontrar uma boa aproximação da *fronteira de Pareto*, como exemplificado na Figura 33.

6.2
Experimentos numéricos para o fJSP

Para resolver o problema fJSP com o método proposto, devem ser aplicados os algoritmos das seções 5.2.1 – 5.2.2.2 – 5.2.3, com as funções objetivo $f_1 = c_M$ =*Makespan*, $f_2 = w_M =$ Carga de trabalho máxima e $f_3 = w_T =$ Carga de trabalho total, assim como o objetivo auxiliar:

$$f_4 = \sum_{m=1}^{N} (\max_{\forall k,i}\{c_{ki}x_{kim}\} - w_m),$$

PUC-Rio - Certificação Digital Nº 1012728/CA

onde f_4 representa o tempo total de máquina parada, calculado como o somatório das diferenças do término de trabalho e a carga de trabalho de todas as máquinas.

As instâncias do problema foram tomadas em Kacem *et al*. (2002a, 2002b), onde cada instância é representada como $K \times N$, como será explicado mais adiante. Os resultados obtidos pelo método proposto nas instâncias (8×8), (10×10) e (15×10) serão comparados com os resultados dos algoritmos: AL+CGA (KACEM *ET AL*.,2002a), PSO+SA (XIA E WU, 2005), GA+VND (GAO *ET AL*., 2008), PSO+TS (ZHANG *ET AL*., 2009) e MOPSO+LS (MOSLEHI & MAHNAM, 2011). Já os resultados das instâncias (4× 5) e (10×7) serão comparados com os resultados dos algoritmos: FL+EA (KACEM *ET AL*., 2002b) e MOPSO+LS (MOSLEHI & MAHNAM, 2011). O algoritmo proposto foi aplicado considerando 50 e 100 soluções iniciais com 10 execuções para cada instância. Não foi possível a comparação de tempos, já que não foram reportados os tempos de execução nos algoritmos comparados.

PUC-Rio - Certificação Digital Nº 1012728/CA

Problema 4×5

Esta é uma instância totalmente flexível, de 4 tarefas e 12 operações, que devem ser processadas em 5 máquinas. A Tabela 13 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

Tabela 13 – Resultados computacionais para a instância 4×5

FL+EA			MOPSO+LS			Método proposto (50)			Método proposto (100)		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
16	34	10	16	32	8	11	32	10	11	32	10
16	35	9	16	33	7	11	34	9	11	34	9
18	32	8				12	32	8	12	32	8
18	33	7				13	33	7	13	33	7

Problema 10×7

Esta também é uma instância totalmente flexível, de 10 tarefas e 29 operações, que devem ser processadas em 7 máquinas. A Tabela 14 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

Tabela 14 – Resultados computacionais para a instância 10×7

FL+EA			MOPSO+LS			Método proposto (50)			Método proposto (100)		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
15	61	11	15	61	11	11	61	11	11	61	11
16	60	12	15	62	10	12	60	12	12	60	12
16	66	10	16	60	12				12	62	10
17	64	10									
18	63	10									

Problema 8×8

Esta é uma instância parcialmente flexível, de 8 tarefas e 27 operações, que devem ser processadas em 8 máquinas. A Tabela 15 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

Tabela 15 – Resultados computacionais para a instância 8×8

AL+CGA			PSO+AS			GA+VND			PSO+TS		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
15	79	-	15	75	12	14	77	12	14	77	12
16	75	-	16	73	13				15	75	12

MOPSO+LS			Método proposto (50)			Método proposto (100)		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
14	77	12	15	75	12	14	77	12
15	75	12	16	73	13	15	75	12
16	73	13	16	77	11	16	73	13
16	78	11				16	77	11
17	77	11						

Problema 10×10

Esta é uma instância totalmente flexível, de 10 tarefas e 30 operações, que devem ser processadas em 10 máquinas. A Tabela 16 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 16 – Resultados computacionais para a instância 10×10

AL+CGA			PSO+AS			GA+VND			PSO+TS		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
7	45	5	7	44	6	7	43	5	7	43	6

MOPSO+LS			Método proposto (50)			Método proposto (100)		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
7	42	6	7	42	6	7	42	6
7	43	5	8	41	7	8	41	7
8	41	7	8	42	5	8	42	5
8	42	5						

Problema 15×10

Esta é uma instância totalmente flexível, de 15 tarefas e 56 operações, que devem ser processadas em 10 máquinas. A Tabela 17 mostra os resultados obtidos pelo método proposto e outros métodos existentes.

Tabela 17 – Resultados computacionais para a instância 15×10

AL+CGA			PSO+AS			GA+VND			PSO+TS		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
23	95	11	12	91	11	11	91	11	11	93	11
24	91	11									

MOPSO+LS			Método proposto (50)			Método proposto (100)		
c_M	w_M	w_T	c_M	w_M	w_T	c_M	w_M	w_T
11	91	11	13	92	12	12	92	12
12	93	10	13	93	11	13	91	12
			14	92	11	13	92	11

Os resultados dos experimentos numéricos para o fJSP indicam que o método proposto, para todas as instâncias testadas, conseguiu obter todas as soluções *não-dominadas*, com 50 e 100 soluções iniciais, quando comparado com os algoritmos existentes. Para a instância 15×10, com 56 operações e 10

PUC-Rio - Certificação Digital Nº 1012728/CA

máquinas, se obteve uma boa aproximação das soluções *não-dominadas*. Ressalte-se que nas instâncias 4×5 e 10×7 os resultados obtidos são melhores que todos os algoritmos existentes mencionados.

6.3
Experimentos numéricos para o iRS/OS

Para resolver o problema iRS/OS usando o método proposto, foram aplicados os algoritmos das seções 5.2.1 – 5.2.2.1 – 5.2.3, com as funções objetivo $f_1 = c_M$ =*Makespan* e $f_2 = w_B =$ Balanceamento de carga de trabalho, assim como o mesmo objetivo auxiliar $f_4 =$ tempo total de máquina parada, especificado no problema fJSP da seção anterior.

As instâncias do problema foram tomadas em Moon *et al*. (2002b). Uma instância do problema iRS/OS é representada por $K \times J \times N$, como será explicado mais adiante. Os melhores resultados obtidos pelo método proposto serão comparados com a abordagem de Moon, Kim e Gen (2004b) e o método moGA (ZHANG *ET AL*., 2006b), considerando a minimização do *makespan* e o balanceamento da carga de trabalho. O algoritmo proposto foi aplicado considerando 50 e 100 soluções iniciais com 10 execuções para cada instância. Não foi possível a comparação de tempos, já que não foram reportados os tempos de execução nos algoritmos comparados.

Problema 5×21×5

Esta é uma instância parcialmente flexível, de 5 pedidos com um total de 21 operações, que devem ser processadas em 5 máquinas. A Tabela 18 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 18 – Resultados computacionais para a instância 5×21×5

Abordagem Moon, Kim & Gen		moGa		Método Proposto (50)		Método Proposto (100)	
c_M	w_B	c_M	w_B	c_M	w_B	c_M	w_B
1.500	$1{,}57\times10^4$	1.500	$1{,}57\times10^4$	1.490	3.400	1.490	3.400
				1.546	3.144	1.500	3.144
				2.083	1.864	1.651	1.864

Problema 7×30×5

Esta é uma instância parcialmente flexível, de 7 pedidos com um total de 30 operações, que devem ser processadas em 5 máquinas. A Tabela 19 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

Tabela 19 – Resultados computacionais para a instância 7×30×5

Abordagem Moon, Kim & Gen		moGa		Método Proposto (50)		Método Proposto (100)	
c_M	w_B	c_M	w_B	c_M	w_B	c_M	w_B
1.965	$1{,}49\times10^4$	1.894	$1{,}21\times10^4$	1.800	1.144	1.800	1.144
				1.820	480	1.820	480
				1.955	264	1.955	264

Problema 10×43×5

Esta é uma instância parcialmente flexível, de 10 pedidos com um total de 43 operações, que devem ser processadas em 5 máquinas. A Tabela 20 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

PUC-Rio - Certificação Digital Nº 1012728/CA

Tabela 20 – Resultados computacionais para a instância 10×43×5

Abordagem Moon, Kim & Gen		moGa		Método Proposto (50)		Método Proposto (100)	
c_M	w_B	c_M	w_B	c_M	w_B	c_M	w_B
2.592	$1{,}94\times10^4$	2.487	$1{,}89\times10^4$	2060	1256	2060	1256
				2140	544	2140	544
				2277	456	2277	456
				2337	320	2337	320

Os resultados dos experimentos numéricos para o iRS/OS indicam que o método proposto conseguiu melhores resultados em todas as instâncias com 50 e 100 soluções iniciais, quando comparado com os algoritmos existentes.

6.4
Experimentos numéricos para o APS

Para resolver o problema APS usando o método proposto, foram aplicadas as mesmas especificações do problema iRS/OS da seção anterior.

Na literatura somente encontrou-se uma única instância para este problema em Moon *et al*. (2004b). As instâncias testes de um problema APS não são publicadas, possivelmente porque seus dados numéricos são gerados aleatoriamente. Uma instância do problema APS é representada por $K \times J \times D \times N$, como será explicado mais adiante. Os melhores resultados obtidos pelo método proposto são comparados com a abordagem de Moon Kim e Gen (2004b), o algoritmo AGA (MOON *ET AL*., 2006) e o método moGA (ZHANG *ET AL*., 2006a). Eles resolveram este problema considerando unicamente a minimização do *makespan*. O algoritmo proposto será aplicado considerando 50 e 100 soluções iniciais com 10 execuções para verificar a eficácia do algoritmo. Mais uma vez, não foi possível a comparação de tempos, já que não foram reportados os tempos de execução nos algoritmos comparados.

Problema 4×17×2×6

Esta é uma instância parcialmente flexível, de 4 pedidos com um total de 17 operações, com a possibilidade de serem trabalhadas em 2 plantas, que devem ser

PUC-Rio - Certificação Digital Nº 1012728/CA

processadas em 6 máquinas. A Tabela 21 mostra os resultados obtidos pelo método proposto e pelos métodos existentes acima mencionados.

Tabela 21 – Resultados computacionais para a instância 4×17×2×6

Abordagem Moon, Kim & Gen		moGa / AGA		Método Proposto (50)		Método Proposto (100)	
c_M	w_B	c_M	w_B	c_M	w_B	c_M	w_B
1.207	-	1.102	-	1.170	$1{,}92\times10^4$	1.103	$0{,}75\times10^4$
				1.206	$1{,}75\times10^4$	1.207	988,89
				1.267	$1{,}74\times10^4$		
				1.306	$1{,}14\times10^4$		
				1.322	$0{,}75\times10^4$		
				1.347	$0{,}50\times10^4$		
				1.401	$0{,}47\times10^4$		
				1.508	$0{,}34\times10^4$		
				1.539	$0{,}22\times10^4$		
				1.562	$0{,}13\times10^4$		
				1.649	$0{,}11\times10^4$		

Os resultados dos experimentos numéricos com o problema APS indicam que o método proposto obteve boas soluções *não-dominadas* para 50 soluções iniciais. Se for comparado o valor do *makespan* do método proposto com o dos algoritmos existentes, pode-se afirmar que o método proposto esteve perto de atingir o *makespan* de melhor resultado (1.102).

PUC-Rio - Certificação Digital Nº 1012728/CA