(SGS). This SGS generates parameterized active schedules. This step allows to obtain a schedule for each chromosome;

- **Step2**: This step makes use of a local search procedure that attempts to improve the solution obtained previously.

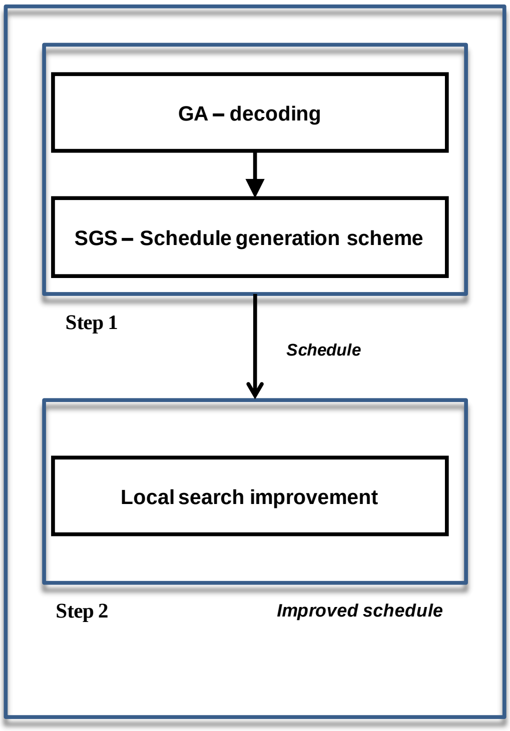

Fig.1 – Architecture of the approach.

The DSS allows the user choose the crossover operator:

- GA-MC-SPc: Single-Point crossover or simple crossover;
- GA-MC-TPc: Two-Point crossover;
- GA-MC-Uc: Uniform crossover or discrete crossover;
- GA-MC-Fc: Flat crossover.

The DSS was developed using the Visual Basic for Applications (VBA) from Microsoft and the Gantt Time Package V3.21 [30].

4 Real-coded evolutionary algorithms

The evolutionary algorithms are an interdisciplinary research area comprising several paradigms inspired by Darwinian principle of evolution.

The current stage of research considers, among others, the following paradigms: genetic algorithms, genetic programming, evolutionary strategies, neuroevolution and differential evolution.

The genetic algorithms have been applied successfully in several areas, such as bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics and physics.

A real-coded GA is adopted in this article. Compared with the binary-code GA, the real-coded GA has several distinct advantages, which can be summarized as follows, Y.-Z. Luo et al. [36]:

- It is more convenient for the real-coded GA to denote large scale numbers and search in large scope, and thus the computation complexity is amended and the computation efficiency is improved;
- The solution precision of the real-coded GA is much higher than that of the binary-coded GA;
- As the design variables are coded by floating numbers in classical optimization algorithms, the real-coded GA is more convenient for combination with classical optimization algorithms.

5 Genetic Algorithms

Genetic algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [1].

The general schema of GA may be illustrated as follows (Fig. 2).

```
procedure GENETIC-ALGORITHM

Generate initial population P0;
 Evaluate population P0;
 Initialize generation counter g ←0;

 While stopping criteria not satisfied repeat
    Select some elements from Pg to copy into Pg+1;
    Crossover some elements of Pg and put into Pg+1;
    Mutate some elements of Pg and put into Pg+1;
    Evaluate some elements of Pg and put into Pg+1;
    Increment generation counter: g ← g+1;
 End while

End GENETIC-ALGORITHM;
```

Fig.2 - Pseudo-code of a genetic algorithm.

First of all, an initial population of potential solutions (individuals) is generated randomly. A selection procedure based on a fitness function enables to choose the individuals candidate for reproduction. The reproduction consists in recombining two individuals by the crossover

Lawrence [14] and ORB from Applegate and Cook [29].

Problem	*BKS*	*GA-MC-SPc*	*GA-MC-TPc*	*GA-MC-Uc*	*GA-MC-Fc*
LA01	666	666	666	666	666
LA02	655	655	655	655	655
LA03	597	597	597	597	597
LA04	590	590	590	590	590
LA05	593	593	593	593	593
LA06	926	926	926	926	926
LA07	890	890	890	890	890
LA08	863	863	863	863	863
LA09	951	951	951	951	951
LA10	958	958	958	958	958
LA11	1222	1222	1222	1222	1222
LA12	1039	1039	1039	1039	1039
LA13	1150	1150	1150	1150	1150
LA14	1292	1292	1292	1292	1292
LA15	1207	1207	1207	1207	1207
LA16	945	945	946	946	946
LA17	784	784	784	784	784
LA18	848	848	848	848	848
LA19	842	842	842	842	842
LA20	902	902	902	902	902
LA21	1046	1054	1052	1058	1059
LA22	927	932	935	932	935
LA23	1032	1032	1032	1032	1032
LA24	935	944	949	944	955
LA25	977	985	984	989	993
LA26	1218	1218	1218	1218	1218
LA27	1235	1252	1252	1266	1273
LA28	1216	1231	1234	1228	1248
LA29	1157	1180	1213	1193	1209
LA30	1355	1355	1355	1355	1355
LA31	1784	1784	1784	1784	1784
LA32	1850	1850	1850	1850	1850
LA33	1719	1719	1719	1719	1719
LA34	1721	1721	1721	1721	1721
LA35	1888	1888	1888	1888	1888
LA36	1268	1278	1282	1284	1284
LA37	1397	1408	1411	1422	1407
LA38	1196	1213	1219	1232	1247
LA39	1233	1244	1250	1250	1249
LA40	1222	1233	1246	1233	1245
ABZ5	1234	1234	1234	1234	1234
ABZ6	943	943	943	943	943
ABZ7	656	688	695	690	693
ABZ8	645	704	708	702	701
ABZ9	661	715	720	724	721
FT06	55	55	55	55	55
FT10	930	930	930	930	930
FT20	1165	1165	1165	1165	1165
ORB1	1059	1059	1059	1059	1059
ORB2	888	888	888	888	888
ORB3	1005	1005	1021	1020	1020
ORB4	1005	1005	1011	1011	1011
ORB5	887	887	889	889	889
ORB6	1010	1013	1013	1013	1013
ORB7	397	397	397	397	397
ORB8	899	899	899	899	899
ORB9	934	934	934	934	934
ORB10	944	944	944	944	944
% ARD		0,60%	0,80%	0,78%	0,89%

Table 6: Average relative deviation with all problems.

As different authors used different number of problems, the comparison is based only on those problems that authors considered. Note that the authors have different approaches to solve the JSSP problem.

We compare the obtained results with the aforementioned state of art approaches:

- Qing-dao-er-ji and Wang [38];
- Rego and Duarte [8];
- Hasan et al. [27];
- Gonçalves et al. [4];
- Aarts et al. [15];
- Dorndorf et al. [18];
- Binato et al. [20];
- Nowicki and Smutnicki [31];
- Sha and Hsu [6];
- Pardalos et al. [32];
- Adams et al. [10];
- Zhang et al. [35].

The result is showed in Table 7.

The algorithm GA-MC-SPc gives good results when compared to the state of art. In Table 7 we can see that the algorithm proposed is among the top ten best performing.

This study limited the number of generations for all operators tested. The number of generations was limited to 400. A higher number of generations for all operators would increase the computational time. Although there is convergence of the algorithms close to 400 generations, an increase of generations in the proposed algorithm can improve its global performance.

The experiments were performed on an Intel Core 2 Duo CPU T7250 @2.00 GHz. The computational times dispended are in the range [7, 3100] seconds.

NIS	*Test problems*	*Authors*	*Algorithm*	*ARD(%)*
		This paper	GA-MC-SPc	0.31
		Qing-dao-er-ji and Wang [38]	HGA(1)	0.18
40	LA01-LA40	Rego and Duarte [8]	F&F-PRD	0.29
		Hasan et al. [27]	GR-SA-RA	0.97
		Gonçalves et al. [4]	P. Active	0.41
		Gonçalves et al. [4]	Non-Delay	1.20
		Gonçalves et al. [4]	Active	1.10
		Aarts et al. [15]	GLS1	2.05
		Aarts et al. [15]	GLS2	1.75
		Dorndorf et al. [18]	PGA	4.00
		Dorndorf et al. [18]	SBGA	1.25
		Binato et al. [20]	-	1.87
		Nowicki and Smutnicki [31]	i-TSAB	0.05
		Sha and Hsu [6]	HPSO	0.02
		Sha and Hsu [6]	PSO	0.37
		Pardalos et al. [32]	GES	0.00
		Adams et al. [10]	SBI	3.67
		This paper	GA-MC-SPc	0.03
10	ORB1-ORB10	Nowicki and Smutnicki [31]	i-TSAB	0.00
		Zhang et al. [35]	TS	0.00
		Pardalos et al. [32]	GES	0.00

Table 7: Comparison of the % deviations for the different number of problems authors considered.

7 Conclusions and remarks

This paper presents an experimental study of a set of crossover operators, namely single-crossover, two-point crossover, uniform crossover and flat crossover. The single-crossover has the best results, followed by the uniform crossover operator.

This can be a very important result in choosing the crossover operator when using genetic algorithms, particularly in job shop.

Additionally, the best crossover operator was tested on a set of 50 standard instances taken from the literature and compared with the best state-of-the-art approaches. The algorithm produced good results when compared with other approaches.

Although there is convergence of the algorithms close to 400 generations, an increase of generations in the proposed algorithm (GA-MC-SPc) can improve its global performance.

Further work could be conducted to explore the possibility of genetically correct the chromosomes supplied by the genetic algorithm to reflect the solutions obtained by the local search heuristic.

Acknowledgements

This work has been partially supported by the CIDEM - Centre for Research & Development in Mechanical Engineering. CIDEM is a unit of FCT – Portuguese Foundation for the Science and Technology.

References:

[1] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.

[2] J. Magalhães-Mendes, Project scheduling under multiple resources constraints using a genetic algorithm, *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, Issue 11, Volume 5, November 2008, pp. 487-496.

[3] I. Essafi, Y. Mati and S.D. Pérès, A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem, *Computers & Operations Research*, Vol. 35, Issue 8, 2008, pp. 2599-2616.

[4] J.F. Gonçalves, J.M. Mendes, and M.C.G. Resende. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, Vol. 167, 2005, pp. 77-95.

[5] J.J.M. Mendes, J.F. Gonçalves and M.G.C. Resende, A random key based genetic algorithm for the resource constrained project scheduling problem. *Comput. Oper. Res.* 36, 1, 2009, pp. 92-109.

[6] D. Y. Sha and C. Hsu, A hybrid particle swarm optimization for job shop scheduling problem. *Computers & Industrial Engineering*, 51, 4, 2006, pp. 791-808.

[7] F. Pezzela and E. Merelli, A tabu search method guided by shifting bottleneck for the job shop scheduling problem, *European Journal of Operational Research*, Vol. 120, 2000, pp. 297-310.

[8] C. Rego and R. Duarte, A filter-and-fan approach to the job shop scheduling problem, *European Journal of Operational Research*, Vol. 194, 2009, pp. 650–662.

[9] E. Nowicki and C. Smutnicki, A Fast Taboo Search Algorithm for the Job-Shop Problem, *Management Science*, Vol. 42, No. 6, 1996, pp. 797-813.

[10] J. Adams, E. Balas and Z. Zawack, *The shifting bottleneck procedure for job shop scheduling*, Management Science, Vol. 34, 1988, pp. 391-401.

[11] C. Y. Zhang, P. Li. and Z. Guan, A very fast TS/SA algorithm for the job shop scheduling problem, *Computers & Operations Research*, Vol. 35, 2008, pp. 282-294.

[12] H.R. Lourenço, Local optimization and the job-shop scheduling problem, *European Journal of Operational Research*, Vol. 83, 1995, pp. 347-364.

[13] H. Fisher and G.L. Thompson, *Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules*, in: Industrial Scheduling, J.F. Muth and G.L. Thompson (eds.), Prentice-Hall, Englewood Cliffs, NJ, 1963, pp. 225-251.

[14] S. Lawrence, *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques*, GSIA, Carnegie Mellon University, Pittsburgh, PA, 1984.

[15] E.H.L.Aarts, P.J.M.Van Laarhoven, J.K. Lenstra and N.L.J.Ulder, A computational study of local search algorithms for job shop scheduling, *ORSA Journal on Computing*, 6, 1994, pp. 118-125.

[16] R.M.Aiex, S.Binato and M.G.C. Resende, Parallel GRASP with Path-Relinking for Job Shop Scheduling, *Parallel Computing*, Vol. 29, Issue 4, 2003, pp. 393 - 430.

[17] F. Croce, R. Tadei, and G. Volta, A Genetic Algorithm for the Job Shop Problem, *Computers and Operations Research*, Vol. 22(1), 1995, pp. 15-24.

[18] U. Dorndorf, and E. Pesch, Evolution Based Learning in a Job Shop Environment, *Computers and Operations Research*, Vol. 22, 1995, pp. 25-40.

[19] L. Wang, and D. Zheng, An effective hybrid optimisation strategy for job-shop scheduling problems, *Computers & Operations Research*, Vol. 28, 2001, pp. 585-596.

[20] S. Binato, W.J.Hery, D.M. Loewenstern and M.G.C.Resende, *A GRASP for Job Shop Scheduling*. In: Essays and Surveys in Metaheuristics, Ribeiro, Celso C., Hansen, Pierre (Eds.), Kluwer Academic Publishers, 2002.

[21] B. Giffler and G.L. Thompson, Algorithms for Solving Production Scheduling Problems, *Operations Research*, Vol. 8(4), 1960, pp. 487-503.

[22] J. Carlier and E. Pinson, An Algorithm for Solving the Job Shop Problem. *Management Science*, Feb, 35(29), 1989, pp.164-176.

[23]J. Carlier and E. Pinson, A practical use of Jackson's preemptive schedule for solving the job-shop problem. *Annals of Operations Research*, Vol. 26, 1990, pp. 269-287.

[24]P. Brucker, B. Jurisch and B. Sievers, A Branch and Bound Algorithm for Job-Shop Scheduling Problem, *Discrete Applied Mathematics*, Vol. 49, 1994, pp. 105-127.

[25]D. P.Williamson, L.A. Hall, J.A. Hoogeveen, , C. A. J.Hurkens, J. K. Lenstra, S. V. Sevastjanov and D. B. Shmoys, Short Shop Schedules, *Operations Research*, 45(2), 1997, pp. 288-294.

[26]J. Magalhães-Mendes, Project scheduling using a competitive genetic algorithm. In *Proceedings of the 8th WSEAS Conference on Simulation, Modelling and Optimization* (Santander, Cantabria, Spain). J. M. de la Maza and P. L. Espí, Eds. Mathematics And Computers In Science And Engineering. WSEAS, Stevens Point, Wisconsin, 2008, pp. 39-42.

[27]S.M.K. Hasan, R. Sarker and D. Cornforth, GA with Priority Rules for Solving Job-Shop Scheduling Problems. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC(2008)*, 2008, pp. 1913-1920.

[28]M. J. Varnamkhasti, L. S. Lee, M. R. Bakar, and W. J. Leong, A Genetic Algorithm with Fuzzy Crossover Operator and Probability, *Advances in Operations Research*, vol. 2012, Article ID 956498, 16 pages, 2012. doi:10.1155/2012/956498

[29]D. Applegate and W. Cook, A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2), 1991, pp. 149-156.

[30]J. Magalhães-Mendes, A genetic algorithm for the job shop scheduling with a new local search using Monte Carlo method, In *Proceedings of the 10th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases* (AIKED'11), Cambridge, UK, 2011, pp. 26-31.

[31]E. Nowicki and C. Smutnicki, An Advanced Tabu Search Algorithm for the Job Shop Problem, *Journal of Scheduling*, 8, 2005, pp.145–159.

[32]P. Pardalos, O. Shylo and A. Vazacopoulos, Solving job shop scheduling problems utilizing the properties of backbone and big valley, *Computational Optimization and Applications*, 47(1), 2010, pp.1-16.

[33]G. Sywerda, Uniform crossover in genetic algorithms, in Proceedings of the *3rd International Conference on Genetic Algorithms*, 1989, pp. 2-9.

[34]K. A. De Jong and W. M. Spears, A formal analysis of the role of multi-point crossover in genetic algorithms, *Annals of Mathematics and Artificial Intelligence*, vol. 5, no. 1, 1992, pp. 1–26.

[35]C.Y. Zhang, P.G. Li, Z.L. Guan, and Y.Q. Rao. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research*, 34(11), 2007, pp. 3229-3242.

[36]Y.-Z. Luo, G.-J. Tang, Z.G. Wang and H.Y. Li. Optimization of perturbed and constrained fuel-optimal impulsive rendezvous using a hybrid approach. *Engineering Optimization*, 38(8), 2006, pp. 959-973.

[37]R. R. Sharapov. *Genetic Algorithms: Basic Ideas, Variants and Analysis*, Vision Systems: Segmentation and Pattern Recognition, Goro Obinata and Ashish Dutta (Ed.), 2007, pp. 407-422, ISBN: 978-3-902613-05-9.

[38]R. Qing-doa-er-ji and Y. Wang. A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, 39(10), 2012, pp. 2291-2299.

[39]H.R. Choi and B.J. Park, Genetic Algorithm for the Integration of Process Planning and Scheduling in a Job Shop, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, Issue 12, Volume 3, December 2006, pp. 2498-2504.

[40]H. Chiu, K. Hsieh, Y.T. Tang and W. Chien, Employing a Genetic Algorithm Based on Knowledge to Address the Job Shop Scheduling Problem, *WSEAS TRANSACTIONS on COMPUTER RESEARCH*, Issue 2, Volume 2, February 2007, pp. 327-333.

[41]H. Chiu, K. Hsieh, Y.T. Tang and C.Y. Wang, A Novel Approach to Address the Job-Shop Scheduling Problem by using a Tabu Genetic Algorithm, *WSEAS TRANSACTIONS on COMPUTER RESEARCH*, Issue 2, Volume 2, February 2007, pp. 339-345.