The Job Shop Scheduling Problem Solved with the Travel Salesman Problem and Genetic Algorithms

INFORMS Annual Meeting 2013
Eva Selene Hernández Gress
Autonomous University of Hidalgo

Abstract

In this paper we proposed a solution to the JobShop Scheduling Problem using the Traveling Salesman Problem solved by Genetic Algorithms. Different tests are performed to solve the Traveling Salesman Problem with the two types of selection (tournament and roulette) under different parameters: number of individuals, number of iterations, crossover probability and mutation probability. Then the best type of selection and the best parameters are used to solve the Job-Shop Scheduling Problem through the Traveling Salesman Problem. Different cases in the literature are solved to compare results.

Agenda

- Introduction
- The Travel Salesman Problem solved with Genetic Algorithms
- The Job Shop Scheduling solved as a Travel Salesman Problem
- Conclusions

Introduction

- This research modeled the Traveling Salesman Problem (TSP) through integer programming to analyze the number of cities that was feasible to solved by this method.
- Then we proposed a Genetic Algorithm which was tested with some examples where the solution was found through integer programming.

Introduction

- Also we used Genetic Algorithms to solve some examples where the solution could not be found by integer programming because the number of constraints grows exponentially as the number of cities visited.

- The TSP solved by genetic algorithms (GA) was used to solve the Job Shop Problem (JSP).

Introduction

- The conventional methods such as integer programming report a border in time to determine the optimal sequence in the JSP in a reasonable computational time (Tamilarasi and Anantha, 2010).
- Through the resolution of TSP with GA a method for solving it is validated.

The Travel Salesman Problem

- The TSP is a combinatorial optimization problem in which a salesman visits only once each of the cities and back to the starting point, the problem consist in locate the path with the shortest distance and it is known as the optimal route.

The Travel Salesman Problem

- The Traveling Salesman has been studied extensively especially with metaheuristics, see for example, the work of Dorigo (1997) with the ant colony method, Cerny (1985) with the Monte Carlo method; Jog et al. (1991) Chatterjee et al. (1996), Larrañaga et al. (2000), Moon et al. (2002), Fogel (2004) etcwith Genetic Algorithms with very good results. William Cook, Vasek Chvátal and Applegate (Applegate, 2006) have solved the problem for 24, 978 cities in 2004.

The Travel Salesman Problem

- The Traveling Salesman Problem consists in choosing the route that minimizes the distance between cities $1, 2, 3, \ldots, N$. For $i \neq j$, C_{ij} is the distance from city i to city j and $C_{ii} = M$, where M is a very large number (relative to the actual distances of the problem).

The Travel Salesman Problem

The following explains how the experiment was performed with the distance matrix proposed by Winston (2005), shown below:

	City 1	**City 2**	**City 3**	**City 4**	**City 5**
City 1	M	132	217	164	58
City 2	132	M	290	201	79
City 3	217	290	M	113	303
City 4	164	201	113	M	196
City 5	58	79	303	196	M

Table 1. Distance matrix (Winston,2005).

The Travel Salesman Problem

(1,5)	58
(2,4)	201
(3,1)	217
(4,3)	113
(5,2)	79

Table 2. Solution with integer programming

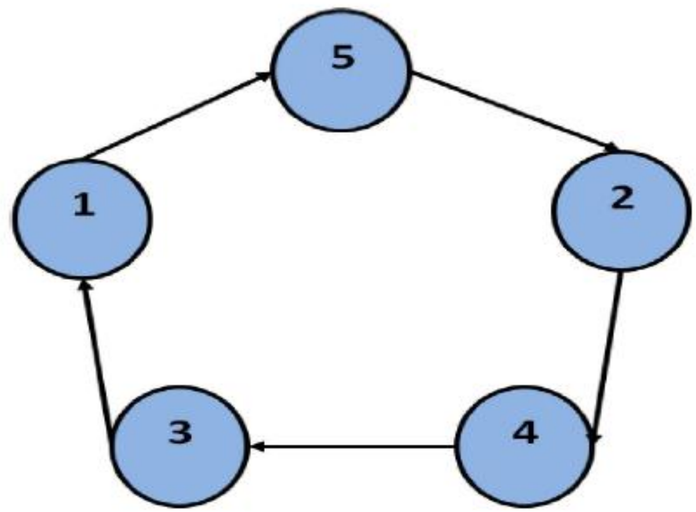


Figure 1. Optimal Route for the Winston (2005) Problem .

Optimal solution=668 units

The Travel Salesman Problem

To begin is required to have a square matrix that represents the cost of the distance to travel from the city *i* to *j*; generating an initial population of a certain number of individuals with random routes, for example, in Table 3 are 10 individuals with five alleles (each allele is a city) and their respective fitness (fitness).

Individual	Route	Cost	Fitness
1	1-2-3-4-5	132+290+113+196+58	789
2	5-3-1-2-4	303+217+132+201+196	1049
3	3-4-5-2-1	113+196+79+132+217	737
4	2-4-5-1-3	201+196+58+217+290	962
5	3-1-4-2-5	217+164+201+79+303	964
6	5-2-3-1-4	79+210+217+164+196	946
7	2-1-5-3-4	132+58+303+113+201	807
8	3-2-1-5-4	290+132+58+196+113	789
9	1-2-5-4-3	132+79+196+113+217	737
10	5-4-1-3-2	196+164+217+290+79	946

Table 3. Random routes

The Travel Salesman Problem

To perform the tournament selection two random permutations of equal size to the number of individuals are generated, for example, P1 = 6-3-7-8-5-1-2-4-9-10 first permutation, the second permutation P2 = 2-4-9-10-6-3-7-8-5-1, 6 and 2 compete and the best (less fitness) is selected. The result can be seen in Table 4.

Competitors	Winner	Route	Fitness
6,2	6	5-2-3-1-4	946
3,4	3	3-4-5-2-1	737
7,9	9	1-2-5-4-3	737
8,10	8	3-2-1-5-4	789
5,6	6	5-2-3-1-4	946
1,3	3	3-4-5-2-1	737
2,7	7	2-1-5-3-4	807
4,8	8	3-2-1-5-4	789
9,5	9	1-2-5-4-3	737
10,1	1	1-2-3-4-5	789

Table 4. Tournament

The Travel Salesman Problem

After ordering the table is performed the crossover. For crossover an arrangement is randomly generated, the rows are the number of individuals between two, the columns are always two. In the example are 5x2 with permutations in each column. Column 1 are the Possible Father 1 and column 2 the Possible Father 2. A crossover probability is generated, for example, 0.6 and a random number for each couple also, if it is less than the crossover probability then, the couple formed by the first line of the column Possible Father 1 and the first element the second column of the Possible Father 2 are selected for the crossover.

Father 1=	5	2	3	1	4
Father 2=	1	2	5	4	3
Child 1=	5	2	5	4	4
Child 2=	1	2	3	1	3

Figure 2. Crossover.

The Travel Salesman Problem

Child 1=	5	2	3	1	4	=Individual 9
Child 2=	1	2	5	4	3	=Individual 1

Figure 3. Crossover, corrected subtours

Individual	Route	Fitness
1	1-2-5-4-3	737
2	4-2-1-5-3	807
3	3-4-5-2-1	737
4	3-4-5-2-1	737
5	3-2-1-5-4	789
6	3-5-2-1-4	791
7	1-2-3-4-5	789
8	2-1-5-3-4	807
9	5-2-3-1-4	946
10	5-2-3-1-4	946

Table 5 .Population after Crossover

The Travel Salesman Problem

The last operator is the mutation, this is done by selecting a probability of mutation, in this case 0.1, and generating a random number for each individual. After generating a random number the only one that turned out to be less than 0.1 was individual 2. For this particular individual, two points are selected for example 2 and 5 position are exchanged.

Individual 2=	4	2	1	5	3
Mutation=	4	3	1	5	2

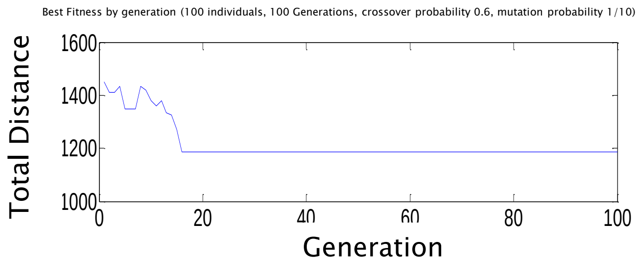
Figure 4. Mutation

The Travel Salesman Problem

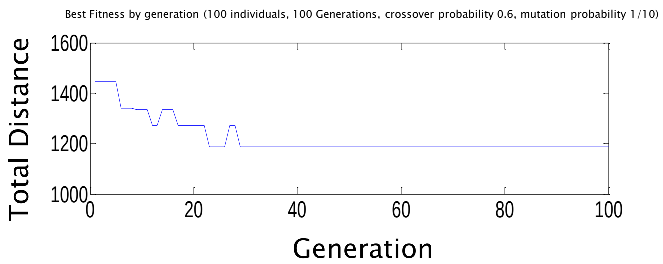
Individual	Route	Fitness
1	1-2-5-4-3	737
2	4-3-1-5-2	668
3	3-4-5-2-1	737
4	3-4-5-2-1	737
5	3-2-1-5-4	789
6	3-5-2-1-4	791
7	1-2-3-4-5	789
8	2-1-5-3-4	807
9	5-2-3-1-4	946
10	5-2-3-1-4	946

Table 6 .Population after Mutation

The Travel Salesman Problem



a)Offline Performance–Tournament Selection



a)Offline Performance–Roulette Selection

Figure 5 . Comparation between a) Tournament Selection b) Roulette Selection

The Travel Salesman Problem

Experiment (Number of Cities)	Selection	Individual	Generation	Crossover probability	Mutation Probability	Time (seconds)	Experiment Result	Best result in literature
(Winston,2005):								
5	Tournament	100	100	0.6	0.1	2	668	668
5	Roulette	100	100	0.6	0.1	2	668	668
(Gerhard,2006):								
10	Tournament	100	100	0.6	1/10	3	1185	1185
10	Roulette	100	100	0.6	1/10	3	1185	1185
10	Tournament	100	100	0.6	0.1	3	7392	8914
10	Roulette	100	100	0.6	0.1	3	7392	8914
15	Tournament	100	100	0.6	0.05	5	1692	1513
15	Roulette	100	100	0.6	0.05	5	1757	1513
20	Tournament	100	100	0.6	1/20	7	1700	1688
20	Roulette	100	100	0.6	1/20	7	1854	1688
21	Tournament	100	100	0.6	0.05	8	2042	2042
21	Roulette	100	100	0.6	0.05	8	2077	2042
280 Cities (Gerhard, 2006)								
AO	Tournament	100	20000	0.6	0.1	420	6213.25	2579
BT	Roulette	100	20000	0.6	0.1	900	7968.55	2579
AP	Tournament	500	20000	0.6	0.1	2220	6750.22	2579
BU	Roulette	1000	20000	0.6	0.1	13680	8686.33	2579
AQ	Tournament	1000	20000	0.6	0.1	4200	5656.54	2579
CA	Roulette	1000	20000	0.6	0.1	13980	8686.33	2579
AR	Tournament	2000	20000	0.6	0.1	8400	6424.03	2579
BX	Roulette	2000	20000	0.6	0.1	28860	10820.19	2579
AS	Tournament	5000	20000	0.6	1/280	46810	5325.67	2579
BI	Roulette	5000	20000	0.6	1/280	76680	7433.69	2579
BZ	Tournament	20000	1000	0.6	1/280	3600	7205.13	2579
AT	Roulette	20000	1000	0.6	1/280	21600	5509.59	2579
BJ	Tournament	50000	1000	0.6	1/280	81140	4724.52	2579

Table 7. Results

From these examples it can be concluded that the tournament selection is the best, it is necessary to consider a relatively large number of individuals in the population with a moderate number of iterations (at a ratio of 10 individuals for one iteration approximately) as we can observe in examples BZ, AT and AJ. The crossover probability works best is the 0.6 and low mutation probability (*1 / n*), where *n* is the number of individuals

The Job Shop Problem

- Among the authors who have used metaheuristics for their solution stand Yamada and Nakano (1998) and Sivanandam and Deepa (2008) with genetic algorithms, Bozejko et al. (2009) with simulated annealing, Huang (2004) and Ge et al. (2007) hybrid algorithms (genetic algorithms and optimization particles) and Anantha & Tamilarasi (2010) with a hybrid genetic algorithm and simulated annealing, etc.

The Job Shop Problem

- The Job Shop Problem is to schedule a set of jobs in a set of machines, subject to the constraint that each machine can handle one job in a time. The objective is to schedule the jobs so as to minimize the maximum of their completition times.

The Job Shop Problem

- This is an exampled presented in Anantha y Tamilarasi (2010):

$O_1=J_1M_1= 2$	$O_2=J_1M_3=3$	$O_3=J_1M_2=4$
$O_4=J_2M_2=1$	$O_5=J_2M_1=5$	$O_6=J_2M_3=2$
$O_7=J_3M_3= 4$	$O_8=J_3M_1=6$	$O_9=J_3M_2=4$

Table 8. JSP with 3 jobs and 3 Machines [4].

The optimal result is 17 units of makespan with their method.

The Job Shop Problem

- To solve the JSP like a TSP, each operation of the JSP is consider as a city of the of the TSP.

Time				
1	2	3	4	5
$O_1=J_lM_l$				
		$O_2=J_1M_3$		

Time				
1	2	3	4	5
$O_1=J_lM_l$				
$O_6=J_2M_3$				

Figure 6: *JSP as a TSP*

The Job Shop Problem

- The distance matrix the TSP is:

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9
O_1	M	5	6	2	7	2	4	8	4
O_2	5	M	7	3	5	5	7	6	4
O_3	6	7	M	5	5	4	4	6	8
O_4	2	3	5	M	6	3	4	6	5
O_5	7	5	5	6	M	7	5	11	5
O_6	2	5	4	3	7	M	6	6	4
O_7	4	7	4	4	5	6	M	10	8
O_8	8	6	6	6	11	6	10	M	10
O_9	4	4	8	5	5	4	8	10	M

Table 9. Distance matrix

The Job Shop Problem

- Solving the problem with GA in Matlab the same result is obtained, the route of operations for the TSP is:

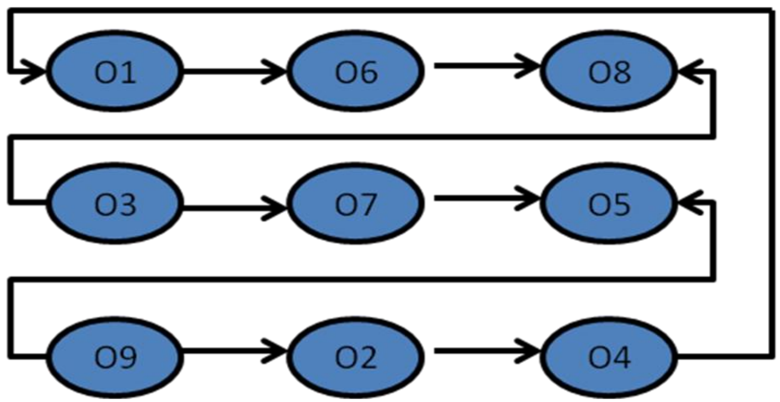


Figure 7. Route for the JSP of Tamilarasi y Anantha (2010).

The Job Shop Problem

- The route can be converted as a JSP:

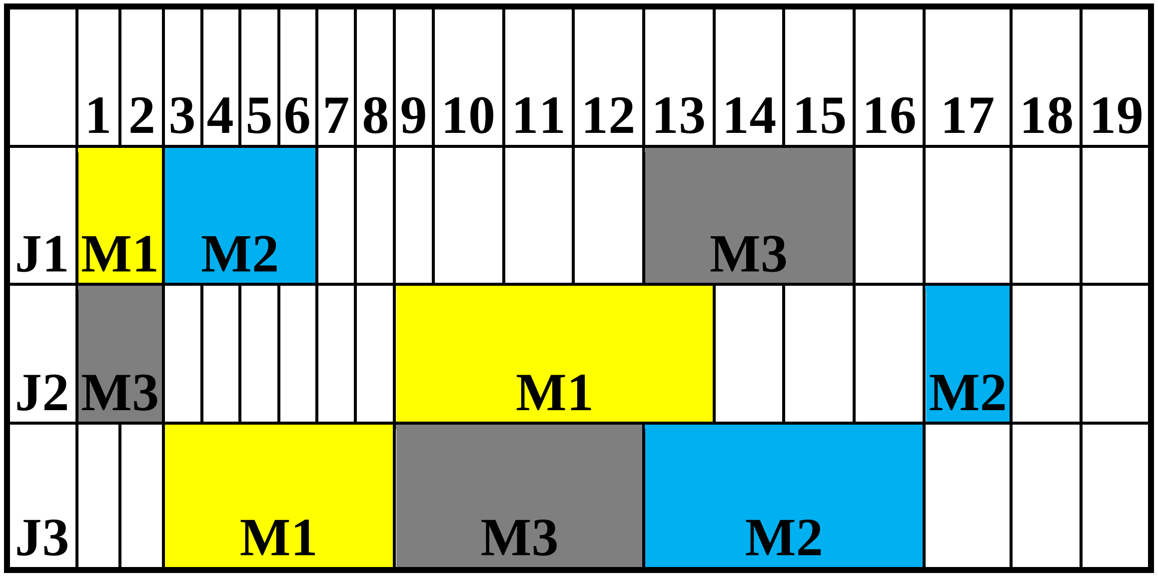


Figure 8. Makespan for the JSSP of Tamilarasi y Anantha (2010)

The Job Shop Problem

- A comparative table is presented with the results of the problems founded in Tamilarasi y Anantha (2010) y Ruiz (2010) and all of them belong to JSP bechmark problems of Beasley (1990).

Experiment	**Individual**	**Generation**	**Crossover probability**	**Mutation Probability**	**Time (seconds)**	**Result obtained in the experiment**	**Best result (Ruiz, 2011)**
Tamilarasi y Anantha (2010)	1000	100	0.6	1/9	120	17	17
FT06 (Ruiz, 2011)	1000	100	0.6	1/36	180	55	55
LA04 (Ruiz, 2011)	60000	300	0.6	1/50	64800	621	590
FT10 (Ruiz, 2011)	50000	1000	0.6	1/100	41230	1156	930
LA02 (Ruiz, 2011)	50000	1000	0.6	1/50	40120	768	655
LA03 (Ruiz, 2011)	50000	1000	0.6	1/50	40100	699	597
LA12 (Ruiz, 2011)	70000	1250	0.6	1/50	110801	1412	1039
LA 13 (Ruiz, 2011)	50000	1000	0.6	1/100	75900	1520	1150

Table 10: Comparative results for the JSP .

Conclusions

The parameters of the GA for solved the TSP that were founded are

- The tournament selection has better performance than the roulette selection.
- The number of individuals is greater than the number of iterations in a proportion of 10 individuals for approximately one iteration.
- Crossover probability shows the best results in 0.6
- The probability of mutation used was relatively low 1 / *n*, where *n* represents the number of cities in the problem.

Conclusions

- Solve the JSP through the TSP is an alternative to address this problem.
- The JSP has been good results solve with metaheuristics.
- In the experiments of JSP, we reach certain results but mostly we just approach the solution found in scientific articles.

References

Applegate, D., 2006. The Traveling Salesman Problem. Princeton University Press, Estados Unidos de América.

Beasley, J., 1990.OR-Library: Distributing test problems by electronic mail. Journal of the Operational Research Society, 11, 1069-1072.

Bozejko, W., Pempera J. and Smuntnicki C., 2009. Parallel simulated annealing for the Job Shop Scheduling problem. Biological Cybernetics, 60, 139-144.

Cerny, V., 1985. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. Lecture note in computer science Proceedings of the 9th Intenational Conference on Computational Science, 5544, 631-640.

Chambers, L., 1998. Genetic Algorithms. University Western Press, Australia.

Chatterjee, S., Carrera C., y Linch L., 1996. Genetic Algorithms and traveling salesman problems. Siam Journal of Optimation, 515-529.

Delgado E., 2005. Aplicación de Algoritmos Genéticos para la Programación de Tareas de una celda de manufactura. Ingeniería e Investigación: Universidad Nacional de Colombia, 24-31.

Dorigo, M., 1997. Ant colonies for the traveling salesman problem. Universidad Libre de Bruselas, Bélgica.

Fogel, D., 1998. An evolutionary approach to the traveling salesman problem. Biological Cybernetics, 60, 139-144.

Ge, H., Du W., y Quian F., 2007. A hybrid algorithm based on swarm optimization and simulated annealing for job shop scheduling. Proceedings of the Third International Conference on Natural Computation, 3, 715-719.

Gerhard, R., 2006. Discrete and Combinatorial Optimization. Universidad de Heidelberg-Instituto de Ciencias de la Computación, Alemania.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Corporation, Estados Unidos de América.

Holland, J., 1992. Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, Estados Unidos de América.

Jog, P., Kim J., Suh J., y Gucht D., 1991. Parallel Genetic Algorithms Applied to the Traveling Salesman Problem. European Journal of Operational Research, 490-510.

Juang, C., 2004. A hybrid genetic algorithm and particle swarm optimization for recurrent network design. IEEE Transactions on Evolutionary Computation, 34, 997-1006.

Larrañaga, P., Kuijpers C., Murga R., Inza I., y Dizdarevic S., 1999. Genetic Algorithms for the Traveling Salesman Problem: A review of Representations and Operators. Artificial Intelligence Review, 129–170.

Moon, C., Kim J., Choi G., y Seo Y., 2002. An efficient genetic algorithm for the traveling salesman problem with precedent constraints. European Journal of Operational Research, 606-617.

Ruiz , J., 2011. Complexity Indicators Applied to the Job Shop Scheduling Problem. International Journal of Combinatorial Optimization Problems and Informatics, 25-31.

Sivanamdam, S. y Deepa S, 2008. Introduction to Genetic Algorithms. Springer, Estados Unidos de América.

Tamilarasi, A. y Anantha K., 2010. An enhanced genetic algorithm with simulated annealing for jobshop scheduling. International Journal of Engineering Science and Technology 2, 144–151.

Winston, W., 2005. Investigación de Operaciones: Aplicaciones y Algoritmos. Indiana University Press, Estados Unidos de América.

Yamada T., y Nakano R., 1997. Genetic Algorithms for Job Shop Scheduling. Proceedings of Modern Heuristic for Decision Support, UNICOM Seminar London, 67-81