A New Estimation of Distribution Algorithm to Solve the Multiple Traveling Salesmen Problem with the Minimization of Total Distance

S.H. Chen[1] and Y.H. Chen[2]

[1] Department of Information Management, Cheng Shiu University. No. 840, Chengcing Rd., Niaosong Dist., Kaohsiung City 83347, Taiwan (R.O.C.)
shchen@csu.edu.tw

[2] Department of Information Science and Applications, Asia University. No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan (R.O.C.)
chenyh@asia.edu.tw

Abstract. Even though the Estimation of Distribution Algorithms (EDAs) have recently been applied to solve many hard problems, only a few EDAs discussed the in-group optimization problems, such as the multiple traveling salesmen problem (mTSP) studied in this research. These problems include the assignment and sequencing procedures in the same time and to be shown in different forms. As a result, this research proposed an algorithm deal by using the Self-Guided GA together with the Minimum Loading Assignment rule (MLA) to tackle the mTSP. We compare the proposed algorithm against the best direct encoding technique, two-part encoding genetic algorithm, in the experiment on the 33 instances drawn from the well-known TSPLIB. The experimental results show the proposed algorithm is better than the compared algorithm in terms of minimization of the total traveling distance. An interesting result also presents the proposed algorithm would not cause longer traveling distance when we increase the number of salesmen from 3 to 10 persons under the objective of minimization of total traveling distance. This research may suggest the EDAs researcher could employ the MLA rule instead of the direct encoding algorithms.

1 Introduction

Estimation of Distribution Algorithms (EDAs) has been discussed extenstively in recent years [4,15,11,13,14]. In particularly, a number of the latest papers on EDAs in solving some NP-hard scheduling problems [12,7,3,9,15,13,10] have shown that EDAs are able to perform effectively. Ceberio et al. [3], in particular, extensively tested 13 famous permutation-based approaches in EDAs on four well-known combinatorial optimization problems. Their paper has provided a good basis for comparison.

Even though EDAs was effective in solving various hard problems, there is a problem that EDA is not discussed extensively. To the best of our knowledge,

J.-S. Pan et al. (eds.), *Intelligent Data Analysis and Its Applications, Volume 1*, Advances in Intelligent Systems and Computing 297,
DOI: 10.1007/978-3-319-07776-5_12, © Springer International Publishing Switzerland 2014

only one EDAs proposed by Shim et al. [14] is able to solve in-group optimization problems, such as the Multiple Traveling Salesmen Problems (mTSP) and the Parallel Machine Scheduling Problems (PMSPs) belonged to this category [1]. In-group optimization problems involve the assignment and routing/sequencing procedures in the same time. Take mTSP for example, a number of n cities are assigned to m salesmen and these n cities are visited once by a salesman where $n > m$. It is apparently that this problem is a NP-Hard problem.

Due to there was only a few EDAs could solve the in-group optimization problems, this research proposed an algorithm deal by using the Self-Guided GA [5] together with the Minimum Loading Assignment rule (MLA) to tackle the mTSP. This strategy is called the transformed-based encoding approach instead of the direct encoding. The solution space of the MLA would be only $n!$. We compare the proposed algorithm against the best direct encoding technique, two-part encoding genetic algorithm (TPGA)[2], in the experimental section. It is notable that solution space of the two-part encoding approach is $n!\binom{n-1}{m-1}$. The proposed method MLA, consequently, is better than the two-part encoding technique. A better solution quality is expected when SGGA works with MLA method.

This rest of the paper is organized as follows: We illustrate the core method of the assignment rule in Section 2 which is applied to Self-Guided GA in Section 3. The experimental results are provided in Section 4 and we draw the conclusions in Section 5.

2 Assignment Rule in the mTSP Problems

Given a set of city sequence $\pi_1, \pi_2, ..., \pi_n$ in π and these cities are not assigned to any salesman yet. This sequence π could be decoded to by assigning the cities to salesmen. That is, the this assignment rule is executed in the fitness function of each chromosome. The rule we called is the minimum loading assignment (MLA) rule. The following pseudo code illustrates the MLA rule.

In the beginning, the first m cities are assigned to the m salesmen and we calculate the objective values of each salesman. The objective function of mTSP would be the total traveling distance or the maximum traveling distance among the salesman. After that, we do the MLA rule iteratively for the unassigned cities. MLA rule assigns the first unassigned city in the sequence π to a salesman when it causes the minimum objective value. This assigned city is removed from the π. This rule is not stopped until there is no city in the π. By using the rule, it means the assigned city could be assigned to a salesman who has the less loading. It also implies that this assigned city might be closed to the last city visited by the salesman so that a far away city would not be considered. Through the MLA rule, it is able to be extended to the parallel machine scheduling problem with setup consideration or the distributed flowshop scheduling problem.

Algorithm 1. Minimum loading assignment rule

Require:
i: The position of a city in the sequence π
$k[i]$: The current number of assigned cities of a salesman i
$\Omega^i_{k[i]}$: The
$i \leftarrow 1$
while $i \leq m$ **do**
  $k[i] \leftarrow 1$
  $\Omega^i_{k[i]} \leftarrow \pi_i$
  $i \leftarrow i + 1$
  $k[i] \leftarrow k[i] + 1$
end while
while $i \leq m$ **do**
  Select a salesman j who could process the π_i with the minimum objective value
  $\Omega^j_{k[j]} \leftarrow \pi_i$
  $i \leftarrow i + 1$
  $k[i] \leftarrow k[i] + 1$
end while

3 Transformed-Based Encoding in Self-Guided Genetic Algorithm

After we introduced the assignment rule in mTSP, this section describes the detail procedures of the Self-guided GA. The benefits of the proposed method are preserving the salient genes of the chromosomes, and exploring and exploiting good searching directions for genetic operators. In addition, since the probabilistic difference provides good neighborhood information, it can serve as a fitness function surrogate. The detailed procedure of the Self-guided GA is described as follows:

Step 1 is the initialization of a population. The sequence of each chromosome is generated randomly.

Step 2 initializes the probability matrix $P(t)$ and the matrix size is $n-by-n$, where n is the problem size. Step 7 builds the probabilistic model $P(t)$ after the selection procedure.In Step 8 and Step 9, $P(t)$ is employed in the self-guided crossover operator and the self-guided mutation operator. The probabilistic model will guide the evolution direction, which is shown in Section 3.2 and Section 3.1. In this research, the two-point central crossover and swap mutation are applied in the crossover and mutation procedures for solving the mTSP under this study.

We explain the proposed algorithm in detail in the following sections. We explain how the probabilistic model guides the crossover and mutation operators.

3.1 Crossover Operator with Probabilistic Model

The idea of Self-Guided Crossover is the same with Self-Guided Mutation, which employs the probability differences of the mating chromosomes by using the

Algorithm 2. MainProcedure of Self-guided GA()

Population: A set of solutions
Generations: The maximum number of generations
$P(t)$: Probabilistic model
t : Generation index

```
1: Initialize Population
2: t ← 0
3: Initialize P(t)
4: while t < generations do
5:    EvaluateFitness (Population)
6:    Selection/Elitism(Population)
7:    P(t+1) ← BuildingProbabilityModel(Selected Chromosomes)
8:    Self-Guided Crossover()
9:    Self-Guided Mutation()
10:   t ← t+1
11: end while
```

Eq. 1. By doing so, we could evaluate which chromosome is mated with a parent solution. For the detail description, please refer in [6].

$$\Delta = \Delta_1 - \Delta_2 = \prod_{p \in (CP1\ to\ CP2), g=[p]}^{n} P(Candidate1_{gp}) - \prod_{p \in (CP1\ to\ CP2), g=[p]}^{n} P(Candidate2_{gp}). \quad (1)$$

3.2 Mutation Operator with Probabilistic Model

Suppose two jobs i and j are randomly selected and they are located in position a and position b, respectively. p_{ia} and p_{jb} denote job i in position a and job j in position b. After these two jobs are swapped, the new probabilities of the two jobs become p_{ib} and p_{ja}. The probability difference Δ_{ij} is calculated as Eq. 2, which is a partial evaluation of the probability difference because the probability sum of the other jobs remains the same.

$$\begin{aligned} \Delta_{ij} &= P(X') - P(X) \\ &\approx \prod_{p \notin (aorb), g=[p]}^{n} P_{t+1}(X_{gp})[(p_{ib}p_{ja}) - (p_{ia}p_{jb})]. \end{aligned} \quad (2)$$

Now that the part of $\prod_{p \notin (aorb), g=[p]}^{n} P_{t+1}(X_{gp})$ is always ≥ 0, it can be subtracted and Eq. 2 is simplified as follows:

$$\Delta_{ij} = (p_{ib}p_{ja}) - (p_{ia}p_{jb}). \quad (3)$$

$$\Delta_{ij} = (p_{ib} + p_{ja}) - (p_{ia} + p_{jb}). \quad (4)$$

If Δ_{ij} is positive, it implies that one gene or both genes might move to a promising area. On the other hand, when Δ_{ij} is negative, the implication is that at least one gene moves to an inferior position.

On the basis of the probabilistic differences, it is natural to consider different choices of swapping points during the mutation procedure. A parameter TM is introduced for the self-guided mutation operator, which denotes the number of tournaments in comparing the probability differences among the TM choices in swap mutation. Basically, $TM \geq 2$ while $TM = 1$ implies that the mutation operator mutates the genes directly without comparing the probability differences among the different TM choices.

When $TM = 2$, suppose the other alternative is that two jobs m and n are located in position c and position d, respectively. The probability difference of exchanging jobs m and n is:

$$\Delta_{mn} = (p_{md} + p_{nc}) - (p_{mc} + p_{nd}). \tag{5}$$

After Δ_{ij} and Δ_{mn} are obtained, the difference between the two alternatives is as follows:

$$\Delta = \Delta_{ij} - \Delta_{mn}. \tag{6}$$

If $\Delta < 0$, the contribution of swapping job m and n is better, so we swap job m and n. Otherwise, jobs i and j are swapped. Consequently, the option of a larger probability difference is selected and the corresponding two jobs are swapped. By observing the probability difference Δ, the self-guided mutation operator exploits the solution space to enhance the solution quality and prevent destroying some dominant genes in a chromosome. Moreover, the main procedure of the self-guided mutation is Eq. 6, where the time-complexity is only a constant after the probabilistic model is employed. This approach proves to work efficiently.

To conclude, the Self-guided GA is obviously different from the previous EDAs. Firstly, the algorithm utilizes the transformed-based encoding instead of using the direct encoding used by Shim et al. [14]. Secondly, the proposed algorithm explicitly samples new solutions without using the crossover and mutation operators. The Self-guided GA embeds the probabilistic model in the crossover and mutation operators to explore and exploit the solution space. Most important of all, the algorithm works more efficiently than previous EDAs [14] in solving the mTSP because the time-complexity is $O(n)$ whereas the previous EDAs needs $O(n^2)$ time.

4 Experimental Results of the Proposed Algorithm

4.1 Experiment Settings

We conducted extensive computational experiments to evaluate the performance of Self-guided GA together with the MLA rule in solving the mTSP. The proposed algorithm was compared with the benchmark encoding algorithm, Two-Part chromosome GA, from the literature [2]. In addition, we employ the genetic operators and parameter settings of Two-Part chromosome genetic algorithm suggested Chen and Chen [8]. The genetic operators are the two-point crossover operator and the swap mutation operator. As a result, it ensures we do a fair comparison between the proposed algorithm with the benchmark encoding algorithm. Besides, a standard genetic algorithm (SGA) also applies the MLA rule which could show the performance enhanced by the assignment rule proposed by this research.

The objective function is to minimize the total traveling distance which is shown in Section 4.2. We implemented the algorithms in Java 2 on a Amazon EC2 with the Windows 2008 server (8-cores CPU). Across all the experiments, we replicated each instance 30 times on the 33 instances from the well-known TSPLIB. We assume the first city of each instance is the home-depot. The size of these instances is from 48 to 400. The number of salesmen is ranging from 2, 3, 5, 10, and 20. As a result, we conduct extensive experiments to evaluate the proposed algorithm under different circumstances.

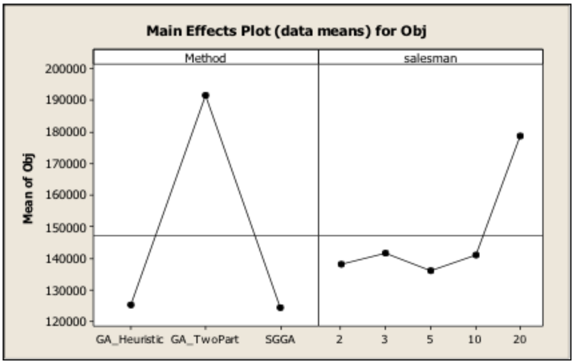

Fig. 1. Main effects plot on the total traveling distance of the compared algorithms

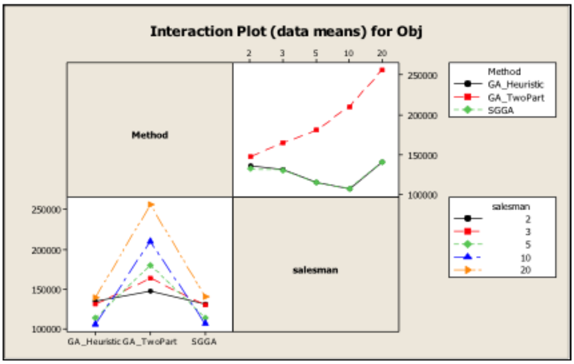

Fig. 2. Intreaction plot on the total traveling distance of the compared algorithms

Table 1. Total Distance

	$GA_{Heuristic}$			$GA_{TwoPart}$			SGGA		
instance	Minimum	Mean	Maximum	Minimum	Mean	Maximum	Minimum	Mean	Maximum
att48	39873	67412	119821	41382	80572	142234	38279	67408	121381
berlin52	9194	13307	20647	9911	15078	23586	8773	13397	21140
bier127	220310	257767	329280	238090	318724	433197	214928	257983	311903
ch130	12781	16173	19262	15896	22951	32313	12020	15959	19035
eil101	1024.4	1251.1	1625.4	1170.4	1734.2	2441.7	910.5	1230.2	1624.1
eil51	497.2	726.6	1195.1	566	896.7	1450.8	483.7	726.2	1186
eli76	771.2	1036.2	1640.5	840.8	1365.6	2151.2	697.2	1030.3	1610.3
gr96	911.3	1325.6	2260.3	1103.9	1834.4	2968.2	893	1325.3	2257.7
kroa150	62801	80353	104886	84155	118336	164472	60914	77752	93631
kroa200	90234	114171	145046	122311	164758	223643	86536	112027	141175
kroB100	39466	52960	74861	49619	78483	118140	39213	52689	78913
kroB150	62532	80549	93700	81609	122203	171691	63013	79185	97548
kroB200	85875	114978	149151	117037	162784	220721	88753	113813	142722
kroC100	37895	53097	77495	55150	81016	128239	37420	51764	78388
kroD100	40864	54795	82342	51506	80103	129610	38858	53964	81408
kroE100	39623	55741	80784	52077	82172	128825	38145	55008	85800
lin105	30255	45563	79901	37418	65266	107157	26922	45533	77804
lin318	182437	243339	322419	251134	328117	447096	183700	243829	318757
mtsp100	40596	54269	81858	50939	80380	123128	39229	54107	83089
mtsp150	76189	96634	116498	101690	138019	184927	76317	94583	118046
mtsp51	467.2	725.2	1188	558.9	897.7	1436.8	475.1	727.6	1198.2
pr124	133373	201905	298646	199047	328870	507277	134543	198651	296597
pr136	223163	282285	383137	273500	419743	634173	200711	281343	390805
pr144	194333	253771	327206	265350	399668	587282	198892	251187	323671
pr152	207562	309617	409260	332865	524397	790537	217451	304029	403995
pr226	329781	519456	753744	588036	842705	1223229	343841	517697	728401
pr264	172047	244558	386775	360147	504250	698832	173869	239184	314209
pr299	218292	283892	361192	314827	413047	585709	220328	282296	357225
pr76	167847	251580	434977	185337	334205	560341	148840	252445	440088
rat195	7200.4	8775.9	10212.1	8563	12055	16873	7059.3	8667.7	10285.1
rat99	2144.1	3418.9	5939.7	2768	4606	7707	2013.9	3392.5	5924
rd400	53411	87481	126827	91499	111470	135460	53204	88769	122815
st70	939.3	1497.3	2594	1142.5	2005.8	3315.9	929.4	1495.7	2641.4
tsp225	13615	16933	20478	17201	23387	32360	13417	16975	20124

4.2 Results of the Total Traveling Distance

This objective evaluates the total distances travelled by the m salesmen. It reflects the total cost of the assignment. Fig. 1 shows the main effects plot on the method comparison and the differences of the number of salesmen we assign. This figure clearly illustrates the SGGA and SGA (denoted $GA_Heuristic$) are better than the Two-Part encoding GA (named $GA_TwoPart$). It means the MLA rule, i.e. the transformed-based method, could be a promising approach which is better than the direct encoding method. Then, when the number of salesmen increased, especially there are 20 salesmen could be assigned, the total distance is increased greatly. As a result, it implies the inefficiency if we request too many salesmen in terms of the managerial perspective.

Fig. 2 depicts the interaction plot between the factor method and the number of salesmen. It might be interesting to see the SGGA and SGA that do not yield the longer total traveling distance when the number of salesmen increased from two to 10 salesmen. However, Two-Part encoding GA may auffer the pain of the number of salesmen increased. This figure could distinguish the effectiveness for the transform-based rule to the direct encoding method. Finally, if a manager would like to determine how many salesmen is required, the lowest total traveling distance would be ten according to this interaction plot.

Finally, the detail result of the three compared algorithms is shown in Table 1.

5 Conclusions

This study solve the in-group optimization problems which is rarely solved by the EDAs. A new EDAs SGGA was proposed, which works with the MLA rule together. In addition, because the MLA rule is classified in the category of transform-based encoding, the proposed algorithm is compared with the two-part encoding GA which is is the best direct encoding strategy so far. We evaluate these algorithm by solving the mTSP problem under 33 instances drawn from TSPLIB. The experimental results show the SGGA with the MLA rule outperforms the Two-Part encoding GA in both the total traveling distance and the maximum traveling objectives. It reveals the proposed algorithm is capable for solving the mTSP problem well. In addition, the MLA rule is also effective and could be applied on some GAs that originally designed for the permutation type problems. As a result, this research provides an insightful results for the researchers who are doing the scheduling problems and could move toward the in-group oprimization problems.

References

1. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34(3), 209–219 (2006)
2. Carter, A.E., Ragsdale, C.T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European Journal of Operational Research 175(1), 246–257 (2006)

3. Ceberio, J., Irurozki, A.M.E., Lozano, J.: A review on Estimation of Distribution Algorithms in Permutation-based Combinatorial Optimization Problems. Accepted by Progress in Artificial Intelligence (2011)
4. Chang, P.C., Hsieh, J.C., Chen, S.H., Lin, J.L., Huang, W.H.: Artificial chromosomes embedded in genetic algorithm for a chip resistor scheduling problem in minimizing the makespan. Expert Systems With Applications 36(3 Pt. 2), 7135–7141 (2009)
5. Chen, S.H., Chang, P.C., Cheng, T.C.E., Zhang, Q.: A self-guided genetic algorithm for permutation flowshop scheduling problems. Computers & Operations Research 39(7), 1450–1457 (2012)
6. Chen, S.H., Chang, P.C., Edwin Cheng, T.C., Zhang, Q.: A self-guided genetic algorithm for permutation flowshop scheduling problems. Computers & Operations Research 39(7), 1450–1457 (2012)
7. Chen, S.H., Chang, P.C., Zhang, Q., Wang, C.B.: A guided memetic algorithm with probabilistic models. International Journal of Innovative Computing, Information and Control 5(12), 4753–4764 (2009)
8. Chen, S.H., Chen, M.C.: Operators of the two-part encoding genetic algorithm in solving the multiple traveling salesmen problem. In: The 2011 Conference on Technologies and Applications of Artificial Intelligence, TAAI 2011 (2011)
9. Chen, S.H., Chen, M.C., Chang, P.C., Chen, Y.M.: Ea/g-ga for single machine scheduling problems with earliness/tardiness costs. Entropy 13(6), 1152–1169 (2011)
10. Chen, S.-H., Chen, M.-C.: Addressing the advantages of using ensemble probabilistic models in estimation of distribution algorithms for scheduling problems. International Journal of Production Economics 141(1), 24–33 (2013)
11. Hauschild, M., Pelikan, M.: An introduction and survey of estimation of distribution algorithms. Accepted by Swarm and Evolutionary Computation (2011)
12. Jarboui, B., Eddaly, M., Siarry, P.: An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. Computers & Operations Research 36(9), 2638–2646 (2009)
13. Pan, Q., Ruiz, R.: An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. Omega 40(2), 166–180 (2012)
14. Shim, V.A., Tan, K., Cheong, C.: A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42(5), 682–691 (2012)
15. Zhang, Y., Li, X.: Estimation of distribution algorithm for permutation flow shops with total flowtime minimization. Computers & Industrial Engineering 60(4), 706–718 (2011)