Extending Population Oriented Extremal Optimisation to Permutation Problems

Marcus Randall

***Abstract*—**Extremal optimisation in its canonical form is based on the manipulation of a single solution. This solution is changed iteratively by gradually replacing poor components of it so that over time it improves. Many successful evolutionary optimisers are population based, so it appears a reasonable exercise to extend extremal optimisation in this way. Scaling it up to an entire population presents many challenges, and only a few works have examined possible models. In this paper, a recent approach is expanded upon which extends the approach from assignment type problems (such as the generalised assignment problem) to permutation oriented ones. Using the asymmetric travelling salesman problem as a test case, it is found that improvements over a canonical extremal optimisation algorithm were realised.

***Keywords*—**Optimisation, Extremal Optimisation, Population, Asymmetric Travelling Salesman Problem.

I. Introduction

Population mechanics are a powerful means to enhance the effectiveness of search in large spaces. The vast majority of evolutionary based algorithms, such as genetic algorithms [1], particle swarm optimisation [2], ant colony optimisation [3] and differential evolution [4] rely on populations of solutions to find either near optimal solutions or good attainment surfaces (in the case of multi-objective optimisation). One relatively recent algorithm, extremal optimisation (EO) [5-7], is an exception to this. It uses a single solution only in its canonical form, and has had relatively modest success compared to the afore mentioned techniques. There have been some recent attempts at developing population structures for the standard algorithm, notably Chen et al. [8-9] and Randall et al. [10-11]. The latter work has demonstrated that suitable algorithmic enhancements could produce improved solutions to assignment type problems (particularly the generalised assignment problem (GAP)) over and above the standard single solution version. In this paper, these ideas are extended to permutation problems, to see if that still holds.

In regard to EO and permutation problems, Randall [12] stated that a number of modifications are necessary in order to produce an EO solver that can effectively explore the different operator state spaces. Additionally, it was noted that there were a number of ways that the solution perturbation could be carried out with local search operators, and these in combination with one another as well. The search power, in fact, was delivered through certain combinations of operators. Therefore, this paper also builds off this work and uses a subset of its test problems for comparative purposes. The technical details such as component ranking strategies and transition endpoint generation are discussed later in this paper.

The remainder of the paper is organised as follows. Section II briefly describes the mechanics of EO and also a recent EO population approach that was successful for solving assignment type problems (like the GAP). Section III develops the concepts of that approach further so that permutation problems can be solved. To test these ideas, Section IV examines the use of a solver based on this design for instances of the asymmetric travelling salesman problem (ATSP). Finally conclusions are presented in Section V.

II. A Brief Description of EO and a Population Approach

Boettcher and Percus [5-7] describe the general tenets of EO. In many ways, it operates counter to other evolutionary optimisation algorithms. Instead of actively seeking good solutions, bad solutions are actively discouraged. The main advantage of this is that EO will not prematurely converge on a locally optimal solution. At each iteration of the algorithm, a poor solution component value (as defined by some incremental cost measure) has its value simply changed to a random value (which, of course, is different from the initial value). In the original version of EO, this chosen solution component was always the worst. However, the performance of EO in this form was not favourable and could not compete with other techniques. To improve this, the component was allowed to be chosen probabilistically by considering the inverse contribution it makes to the quality of the solution. All components are ranked from worst (rank 1) to best (rank n). The set of probabilities are calculated from the ranks. The weighting of these values is dependent on a parameter (conventionally referred to as τ), that allows the search to vary from a random search to a greedy one. The pseudo-code for standard EO is given in Algorithm 1.

Marcus Randall is with the Bond Business School, Bond University, Queensland 4226, Australia (Tel: +61755953361, e-mail: mrandall@bond.edu.au).

Algorithm 1 A generalised algorithm for $\tau-$EO. Note that the termination condition for the overall algorithm is either typically it being run for a defined number of iterations or that a particular solution quality is reached. x is the solution vector.

```
Read problem description and parameters
Form an initial feasible solution to the problem
while the termination condition is not reached do
  Rank the solution components from worst to best
  for $i = 1$ to $n$ do
    $P_i = i^{-\tau}$
  end for
  Normalise $P$
  $i$ = Select component using roulette wheel selection on $P$
  Assign $x_i$ a random (legal) value
  $cost$ = Evaluate the cost of the $x$
  if $cost$ is better than $best\_cost$ then
    $best\_cost = cost$
  end if
end while
Report $best\_cost$
end
```

The important thing to remember is that the above only operates at the single solution level. As mentioned previously, to extend it beyond that, interesting modifications need to be made. Randall and Lewis [13] observed that a population is more than just a collection of individuals, but a set of mechanics which define how the individual solutions interact so as to produce generally better solutions over time. To that end, they developed two possible population models and tested these on an assignment type problem, the GAP. The first was a collective memory scheme in which a social record, across the members of the population, of the performance of solution components is kept and used as part the EO's rules to select the next solution component to change. Its performance on their test problems was unfortunately limited.

The second approach was quite a bit different to this. It relied on a combination of interventions and interactions across the population members instead. At various intervals (either fixed or calculated on a measure of search progress) the process of EO would temporarily stop and the population members would exchange information with one another. The decision of when to temporarily stop was referred to as an *intervention* and the nature of the way the solutions exchanged information at the temporary stop was referred to as an *interaction*. Interaction was achieved through either the elimination of poor solutions (based on dominance criteria) and replacing these with random ones, or the use of a genetic algorithm (GA). In the case of the latter, the GA was run using the members of the archive (as they were dealing with a multi-objective version of the GAP) and its solutions were used to replace poorly performing members in the original population. After either of these interventions, the normal EO process would resume. Interventions would occur several times within an individual deployment of the solver. The overall results revealed that a combination of the GA replacement option and a probabilistic intervention trigger, proved highly effective. In fact, this combination frequently outperformed NSGA-II [14] on many of the larger GAP problem instances.

III. Modifications and Inclusions for Permutation Problems

Given the success of the intervention/interaction population approach of Randall and Lewis [13] as described in Section II, this is adopted as the model used herein. However, a number of adaptations have needed to be made as a result of the permutation structure of the solution vectors and also the fact that there is only one objective to this problem. The work of Randall and Lewis [13] was heavily predicated on the use of multi-objective optimisation artifacts, such as archives and dominance criteria. Both of these adaptations are discussed below.

In the previous work, in order to detect whether an intervention was necessary, they used a measure derived using the number of solutions entering the archive. This was an inverse relationship – the fewer solutions entering the archive (inferring search stagnation), the more likely it would be that an intervention would occur. In the case of a single solution, this no longer applies. While the moderating factor k is retained, a more appropriate relationship may be based on the last time a new best solution is found instead. This is expressed in Equation 1.

$$P = \frac{I-B}{B} \times k \qquad (1)$$

Where:
P is the probability that the interaction will be triggered,
I is the current iteration number,
B is the interaction number that the current best solution was found and
k is the scaling factor introduced by Randall and Lewis [13].

The form of the interaction amongst the population members becomes an interesting issue. As previously mentioned, in the original paper, it was determined that the replacement option using a genetic algorithm performed best overall. For that particular study the generalised assignment problem was used, for which GAs worked particularly well. Given that there are complexities in implementing GAs for permutation problems, and that the aim of this preliminary study is to determine the suitability of population enabled EO to permutation problems, it was determined to simply adapt the solution replacement method proposed by Randall and Lewis [13] instead. This stated that if this type of interaction was undertaken, then the worst solutions would be replaced by randomly generated ones. This had the added benefit that it induced some diversity into the search as well. To allow this to work for single objective problems, a threshold is established – below which solutions are replaced. This is calculated according to Equation 2.

$$T = W - (W - C) \times R \quad (2)$$

Where:
T is the threshold,
W is the objective value of the worst solution in the current population,
C is the objective value of the best solution in the current population and
R is the proportion of the solutions in the population that should be changed – a parameter of the process.

The pseudo-code to perform this update is given in Algorithm 2.

Algorithm 2 Algorithm to update the population based on the replacement strategy.

```
W = Determine the cost of the worst (highest) cost solution in the population
for each $member$ of the population do
    $T = (W - cost(member)) \times R$
    if $cost(member) > T$ then
        Generate random new solution/member
        $cost(member)$ = Evaluate cost of the member according to Equation 3
    end if
end for
end
```

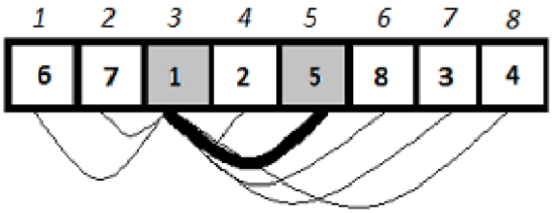
The main questions become which parameter settings (such as population size and intervention particulars) are appropriate and does the technique perform better than canonical EO on permutation problems? These are addressed in Section IV.

A. Permutation Perturbation

The standard EO transition operator as described in Section II is not capable of preserving feasible solutions where a permutation is the solution form. Therefore, Randall [12] modified this mechanism by using the following measures:

- Allowing a variety of standard permutation operators to be applied that work with permutations. These were 1-opt, 2-opt, swap and 3-opt.
- Having two ways for selecting the permutation perturbation endpoint. Each of the operators listed above needs at least another endpoint in the solution to work with. This then ensures a valid permutation is produced. The first option is simply to produce a random endpoint. This second was a neighbourhood approach that examined all possible endpoints and chose the best one. These operators were then distinguished using the prefix *n*. For example, *n*1-opt means that the primary endpoint (the initial city chosen to be moved) is moved to the location that minimises the cost of the transition to the objective. An illustration of this is given in Figure 1.

Fig. 1. x_3 is moved to position 5 so the permutation becomes 6, 7, 2, 5, 1, 8, 3, 4.



In addition, producing a ranked list of the "worst" cities is not intuitive. To do this, Boettcher and Percus' [5-6] "frustration measure" was used. Each city is ranked on the basis of how far it is removed its ideal preceding and proceeding cities. That is, a highly ranked candidate city to change will be well removed from both of these and subsequently have a high probability of being selected according to the mechanics of Algorithm 1.

B. Local Search

Like other evolutionary algorithms, EO needs the use of a subordinate refinement algorithm to move its solution to local optimality. Consistent with other works around the travelling salesman problem (such as Iorache [15], Merz and Freisleben [16] and Stützle, Grün, Linke and Rüttger [17]), a greedy neighbourhood 3-opt local search strategy is used. The procedure is run at each iteration of EO, and terminated once a local optimum has been found. Initial experimentation showed the efficiency of 3-opt with EO over other operators (such as 2-opt). This form of local search was also used by Randall [12].

IV. Computational Experiments

As previously discussed, one of the aims of this paper is to determine the suitability of the population approach on a wider range of problem types, in this case, permutation problems. To that end, the difficult asymmetric travelling salesman problem (ATSP) is used. Its mathematical model is given in Equations 3–4.

$$\text{Minimise } \sum_{i=1}^{n} d(x_i, x_{i+1}) + d(x_n, x_1) \quad (3)$$

s.t.,

$$x_i \neq x_j, \ \forall i, j \quad 1 \leq i, j \leq n \quad i \neq j \quad (4)$$

Where:
n is the number of cities,
$d(i,j)$ is the distance between city i and city j – note that $d(i,j) \neq d(j,i)$ and
x_i is the i^{th} city to be visited on the tour.

The aim of the experiments overall is to determine if adding a population strategy to EO, with appropriate modifications, will produce better results (given the same amount of computational resources) than canonical EO for permutation problems. While simply stated, a large number of experiments will be required. To this end, this paper uses are subset of the test problems presented in Randall [12]. These are given in Table 1 and will allow for comparison with that paper. Note that all experiments are run across ten random seeds for statistical validity. Each run of the solver is allowed up 50,000 EO solution evaluations. This is terminated if the best known cost is reached. Additionally, after initial experimentation a value for *R* was chosen as 0.1 (see Equation 2).

TABLE I
THE ATSP INSTANCES USED IN THIS WORK.

Name	Number of cities (n)	Best Known Cost
ft70	71	38673
ftv70	71	31950
ftv170	171	2755
ftv64	65	1839
kro124p	100	36230

There are two forms of exploration necessary in a study like this, and one may equate these to coarse and fine grain components. These will be referred to as *Phase 1* and *Phase 2* respectively. The experimental setup, and subsequent computational results, for each of these is described below.

A. Phase 1

The fundamental point of this phase is to determine whether using a grouping of solutions in an EO framework is effective for permutation problems. Note the use of the term “grouping” rather than population. At this stage, no population mechanics will be used hence the term *grouping* is preferred. Specifically, no interventions take place in this phase. The aims will be to see both the effects of different grouping sizes and the types of local search transition operators that work with these. The grouping sizes were chosen as {200, 100, 50, 20, 10, 5} which are consistent with those used by Randall [13]. Naturally, in order to ensure a consistent number of overall solutions are evaluated, the number of EO iterations is calculated accordingly. For example, given that 50,000 solutions overall will be evaluated, a grouping size of 200 would mean 250 EO iterations.

The other aspect is the selection of the transition operator probability set. Initial experimentation with the population sizes previously mentioned confirmed that the best setting was the same as found in Randall [12]. Namely the probability set of (0.34, 0.33, 0.33) for the transition operators 1-opt, *n*1-opt and *n*2-opt. This was statistically significantly better regardless of the grouping size.

Table II shows the minimum, median and maximum results for all the identified grouping sizes and the six problem instances. It also gives the canonical EO results. It is clearly evident that the small grouping sizes tend to work better than the larger ones. These results were compared to the original results of Randall [12] using the Kruskal-Wallis test. A statistically significant difference was recorded. A grouping size of 5 received a lower Kruskal-Wallis rank than the original version. This indicates, though does not prove (because of the intervention and interaction mechanisms), that for the next phase, lower population values need further evaluation coupled with the population mechanics.

TABLE II
THE PHASE 1 RESULTS. NOTE THAT ALL VALUES ARE RECORDED AS RELATIVE PERCENTAGE DEVIATIONS (RPDS). THESE ARE CALCULATED AS $\frac{a-b}{b} \times 100$ WHERE a IS THE OBJECTIVE COST VALUE AND b IS THE BEST KNOWN OBJECTIVE COST VALUE (AS DEFINED IN TABLE I). RPDS ARE USED HENCEFORTH TO REPRESENT RESULTS. A POPULATION SIZE OF 1 DENOTES THE ORIGINAL VALUES RECEIVED BY RANDALL [12]. ‘MIN’, ‘MED’ AND ‘MAX’ ARE SHORTENED FORMS OF MINIMUM, MEDIAN AND MAXIMUM RESPECTIVELY. THIS APPLIES AS WELL TO THE FOLLOWING TABLES.

Pop.		Problem				
Size	Measure	ft70	ftv70	ftv170	ftv64	kro124p
200	Min	0	0	3.45	0	0
	Med	0.02	0.36	4.43	0.08	0.27
	Max	0.11	0.56	5.34	0.49	1.06
100	Min	0	0	1.81	0	0
	Med	0.01	0	2.09	0.08	0.03
	Max	0.04	0.21	4.28	0.6	0.31
50	Min	0	0	0.91	0	0.03
	Med	0.05	0.23	2.07	0.16	0.03
	Max	0.14	0.56	2.79	0.6	0.34
20	Min	0	0	0.83	0	0
	Med	0.01	0.13	1.36	0	0.03
	Max	0.17	0.46	3.52	0.49	0.14
10	Min	0	0	0.11	0	0.03
	Med	0.01	0.36	1	0.49	0.03
	Max	0.12	0.56	2.18	0.6	0.03
5	Min	0	0	0.11	0	0
	Med	0.02	0.23	0.74	0.49	0.03
	Max	0.12	0.56	1.38	0.6	0.03
1	Min	0	0	0	0	0
	Med	0	0.36	0.91	0.49	0
	Max	0.15	0.56	4.32	0.6	0.34

B. Phase 2

As indicated previously, this phase will explore the use of various intervention strategies. Again these will be compared against the original results of Randall [12] to answer the question as to whether population mechanics if added to EO will yield a performance difference. Coupled with that are the questions of under what conditions this is possible and why.

Recall that there are two main ways that an intervention could occur. This could either be done regularly (in a pre-defined number of iterations) or probabilistically according to the last time a best solution was received. The values for these are *iterations* = {5000, 10000, 20000} and *s* = {0.2, 0.5, 0.8}. This is consistent with those used by Randall [12]. Note that in a population context, *iterations* refers to the number of solutions that have been evaluated by EO. For example, given that 50,000 solutions are evaluated in total and a population size of 5 is used with an intervention time of 5,000 solutions, at every 1,000th generation of EO an interaction occurs.

It was evident, from Phase 1, that for ATSP and EO, smaller population sizes were the most effective. Hence, for this phase results will be collected for population sizes of 5 and 10, across all of intervention parameters mentioned in the previous paragraph and all the problem instances. These are given in Tables III and IV respectively.

TABLE III
THE PHASE 2 RESULTS USING A POPULATION SIZE OF FIVE MEMBERS. NOTE THAT 'F' INDICATES FIXED WHILE 'P' INDICATES PROBABILITY SETTINGS.

		Problem				
(Type,Value)		ft70	ftv70	ftv170	ftv64	kro124p
(F,5000)	Min	0	0	0	0	0
	Med	0.02	0.11	0.35	0.09	0.04
	Max	0.14	0.47	1.02	0.49	0.04
(F,10000)	Min	0.01	0	0.33	0	0
	Med	0.02	0.36	0.55	0	0.04
	Max	0.12	0.57	2.8	0.6	0.04
(F,20000)	Min	0	0	0.44	0	0
	Med	0.01	0.21	0.48	0	0.04
	Max	0.05	0.47	1.64	0.49	0.04
(P,0.2)	Min	0	0	0	0	0
	Med	0.02	0.36	0.91	0.49	0.04
	Max	0.15	0.57	2.36	0.13	0.04
(P,0.5)	Min	0	0	0.44	0.25	0
	Med	0.01	0.47	1.11	0.6	0.04
	Max	0.21	0.57	2.98	0.49	0.04
(P,0.8)	Min	0	0	0	0.28	0
	Med	0.02	0.21	1.06	0.6	0.04
	Max	0.15	0.57	2.98	0.45	0.04

TABLE IV
THE PHASE 2 RESULTS USING A POPULATION SIZE OF TEN MEMBERS.

		Problem				
(Type,Value)		ft70	ftv70	ftv170	ftv64	kro124p
(F,5000)	Min	0	0	0	0	0.04
	Med	0.01	0.21	0.69	0.33	0.04
	Max	0.14	0.47	1.2	0.6	0.04
(F,10000)	Min	0	0	0.11	0	0
	Med	0	0.24	0.31	0.49	0.04
	Max	0.14	0.57	1.35	0.6	0.04
(F,20000)	Min	0	0	1.09	0	0.04
	Med	0.02	0.24	1.17	0	0.04
	Max	0.21	0.47	1.2	0.49	0.04
(P,0.2)	Min	0	0	0.11	0	0
	Med	0.01	0.21	0.4	0	0.04
	Max	0.11	0.47	1.06	0.17	0.04
(P,0.5)	Min	0	0	0.22	0	0
	Med	0.01	0.11	0.35	0	0
	Max	0.02	0.47	1.02	0.49	0.04
(P,0.8)	Min	0	0	0.11	0	0
	Med	0.01	0.11	0.24	0	0
	Max	0.1	0.57	0.91	0.49	0.04

Visual inspection of Tables III and IV confirms that the smaller population size of five members, on the whole, outperforms EO using ten members for this particular problem at least. In terms of the best combinations for each of the population sizes, somewhat surprisingly, these are different values. Examining the Kruskal-Wallis ranks revealed that for the population size of five, "F,10000" came out as being the best, whereas it was "P,0.5" for size ten. This will require further investigation, but it is suspected that more consistent results may be gained when using a more sophisticated interaction strategy. Beyond this, in comparison to the original results (as reported in Table II), a further test showed that a population of five members, with the above parameters, outperformed the canonical EO. While the differences may appear modest, this then becomes a platform from which to build future versions of population EO.

V. CONCLUSIONS

Populations of solutions within optimisation algorithms are key to providing the search power necessary to solve large and complex problems. Indeed, the more successful evolutionary based algorithms (such as genetic algorithms and particle swarm optimisation) are built around the notion of interacting particles. Extremal optimisation, however, is lot less well known and has only been used across a relatively limited range of problems with more moderate success. Therefore, attempts to improve it by adding population frameworks and mechanics is a desirable thing. This paper has adopted one such framework based on a collection of successful intervention and interaction strategies and adopted these for the use of permutation problems, in particular the ATSP. This is part of a wider effort to extend the reach and utility of this algorithm.

It is evident from all the experimental work in this paper, that having interventions, even with a very simple interaction mechanism, that improvement across performance measures is realisable. This seems particularly evident in the larger problem instances and smaller population sizes (5 members in this case). Interestingly, the problems studied were solved better given a high number of regular interventions. While overall these improvements are currently modest, it is a good indicator that further exploration of these ideas may yield larger improvements.

For the continued development of this framework, it is worthwhile applying it to different types of problems to determine how generally applicable it is. This paper most concentrated on intervention strategies, rather than the different types of interaction that could occur. It is believed, that like the original work, performance would be greatly improved by the diversity that a method like genetic algorithms offers. Therefore, future work will concentrate on determining the most suitable interactions for a range of problem, and importantly if there are commonalities amongst these.

REFERENCES

[1] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA: Addison Wesley, 1989.

[2] J. Kennedy and R. Eberhart, "The particle swam: Social adaptation in social information-processing systems," in *New Ideas in Optimization*. Maidenhead, UK, England: McGraw-Hill, 1999, pp. 379–387.

[3] M. Dorigo and G. Di Caro, "The ant colony optimization metaheuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, 1999, pp. 11–32.

[4] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[5] S. Boettcher and A. Percus, "Extremal optimization: Methods derived from co-evolution," in *Proceedings of the Genetic and Evolutionary and Computation Conference*. Moran Kaufmann, 1999, pp. 825–832.

[6] ——, "Nature's way of optimizing," *Artificial Intelligence*, vol. 119, pp. 275 – 286, 2000.

[7] ——, "Extremal optimization: An evolutionary local search algorithm," in *Computational Modeling and Problem Solving in the Networked World*, ser. Interfaces in Computer Science and Operations Research, H. Bhargava and N. Ye, Eds. Kluwer Academic Publishers, 2003, pp. 61–77.

[8] Y. Chen, Y. Lu, and P. Chen, "Optimization with extremal dynamics for the traveling salesman problem," *Physica A*, vol. 385, pp. 115–123, 2007.

[9] M. Chen, Y. Lu, and G. Yang, "Population-based extremal optimization with adaptive lvy mutation for constrained optimization," in *Computational Intelligence and Security*, ser. Lecture Notes in Computer Science, Y. Wang, Y. Cheung, and H. Liu, Eds. Springer Berlin / Heidelberg, 2007, vol. 4456, pp. 144–155.

[10] M. Randall, "Enhancements to extremal optimisation for generalised assignment," in *The Third Australian Conference on Artificial Life*, ser. Lecture Notes in Artificial Intelligence, M. Randall, H. Abbass, and J. Wiles, Eds., vol. 4828. Berlin: Springer, 2007, pp. 369–380.

[11] M. Randall, T. Hendtlass, and A. Lewis, "Extremal optimisation for assignment type problems," in Biologically-inspired Optimisation Methods: *Parallel Algorithms, Systems and Applications*, ser. Studies in Computational Intelligence, A. Lewis, S. Mostaghim, and M. Randall, Eds. Springer-Verlag, 2009, vol. 210, pp. 139–164.

[12] M. Randall, "Multiple local neighbourhood search for extremal optimisation," in *Proceedings of the Metaheuristics International Conference*, 2013, pp. 278–287.

[13] M. Randall and A. Lewis, "Population extremal optimisation for discrete multi-objective optimisation problems," *Information Sciences*, vol. 367, pp. 390–402, 2016.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[15] Iordache, "Consultant-guided search algorithms with local search for the traveling salesman problem," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature*: Part II, ser. PPSN'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 81–90.

[16] P. Merz and B. Freisleben, "Memetic algorithms for the traveling salesman problem," *Complex Systems*, vol. 13, pp. 297–345, 2001.

[17] T. Stützle, A. Grün, S. Linke, and M. Rüttger, "A comparison of nature inspired heuristics on the traveling salesman problem," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, ser. PPSN VI. London, UK: Springer-Verlag, 2000, pp. 661–670.