Multiple local neighbourhood search for extremal optimisation

Randall, Marcus

Published in:
Proceedings of the X Metaheuristics International Conference

Recommended citation(APA):
Randall, M. (2013). Multiple local neighbourhood search for extremal optimisation. In *Proceedings of the X Metaheuristics International Conference* (pp. 278-287). Singapore Management University.

Multiple Local Neighbourhood Search for Extremal Optimisation

Marcus Randall[1]

Department of Informatics, Bond University
Queensland, Australia, mrandall@bond.edu.au

Abstract

Extremal optimisation (EO) uses a somewhat unusual mechanism to transform one solution into another. This consists of computing a probabilistic worst solution component value, and changing it to a random value. While simple and avoiding problems with premature convergence, it is mostly incompatible with combinatorial problems, particularly those requiring permutations as solution structures. This paper demonstrates that standard local search operators (e.g., 1-opt, 2-opt and 3-opt – used singly or from a neighbourhood) can be readily integrated into the canonical EO framework, without compromising the integrity of the original algorithm. The idea, in some senses may be viewed as a quasi-memetic algorithm. In particular, the primary purpose of this paper is the application and analysis of multiple local search operator neighbourhoods. Issues of solution component ranking techniques and methods for generating local transition operator endpoints are also examined. The difficult and under used asymmetric travelling salesman problem is employed to test these concepts. Results indicate that the simultaneous use of local search operators provides for improved performance over operators used individually.

1 Introduction

Extremal Optimisation (EO) [3, 4] is a relatively under-exploited meta-heuristic that has many desirable properties. One such is lack of convergence – preventing the problem of premature convergence that is a common trait of other techniques [14]. However, this and other characteristics, including ease of implementation and computational simplicity, come at a cost. A non-standard transition operator, in which a probabilistic worst solution component at each iteration is changed to a random value, is its way of traversing state space. Unfortunately, this approach is incompatible with well-known and effective local search operators such as 1-opt, 2-opt and 3-opt. These operators, however, are commonly used as part of many powerful search techniques. Used in the way proposed in this paper, the revised EO algorithm approaches the concept of memetic algorithms [12] which often combine local search heuristics in the one implementation. The purpose of this paper, therefore, is to demonstrate that local search transitions can be sensibly incorporated into the EO framework with minimal change to the standard algorithm. Moreover, multiple local search operator neighbourhoods are investigated to determine if a performance advantage can be leveraged. In addition to this, operator endpoint generation and solution component ranking techniques are examined. The more difficult version of the standard travelling salesman problem (TSP), the asymmetric travelling salesman problem (ATSP) [5], is used to test these ideas.

The remainder of the paper is organised as follows. Section 2 gives a brief description of the mechanics of EO as well as reviewing two EO implementations solving travelling salesman problems. Section 3 describes how local search transition operators can be seamlessly integrated into the EO framework using a probabilistic selection model for multiple neighbourhoods. Related issues of ranking strategies and transition endpoint generation are also discussed. Experimental work is carried out on these aspects in Section 4. Importantly, an investigation and analysis of the effect of transition operators is given using a method labelled as hierarchical parameter tuning. The final computational results indicate operators used in combination often produce better quality solutions than operators used by themselves for the test problem. Finally, conclusions and ideas for further investigation are given in Section 5.

2   EO and the Travelling Salesman

While this paper deals with a significant variation to the standard algorithm, it is important to have a base understanding of its mechanics. More in-depth descriptions may be found in Boettcher and Percus [3, 4] and Randall, Hendtlass and Lewis [15].

Extremal optimisation is an evolutionary meta-heuristic algorithm, that in its canonical form, manipulates a single solution vector. EO alters its solution iteratively, and as such, requires an initial solution. At each iteration, EO will select one of the worst elements[1] of the vector to have its value changed to a random value. To make this selection, the elements are first ranked from worst (rank 1) to best (rank n, where n is the length of the vector), and probabilities assigned, according to the distribution $P_i \propto i^{-\tau}, 1 \leq i \leq n$, where i is the rank. Values of τ close to, or equal to, zero produce an undirected random search strategy. Conversely, allowing $\tau = \infty$ ensures the worst element is chosen each time. Algorithm 1 shows the pseudocode for a single iteration of the EO algorithm.

Algorithm 1 A single EO iteration.

Rank the solution components in x from worst to best according to their adverse effect on the objective function $f(x)$
$i =$ Select a component from x using roulette wheel selection on P
Assign x_i a random (legal) value
$c =$ Evaluate $f(x)$
if c is better than $Best$ **then**
    $Best = c$
end if
end

Boettcher and Percus' [3] solve the standard TSP using a "frustration" measure to determine the rank of each city. This is described in detail in Section 3.2 and forms an integral part of this study. Essentially, however, the ranking reflects how far away a city is from its (ideal) two closest neighbours. Highly frustrated cities are likely to be chosen to be changed. This change is essentially an elaborate 2-opt operation. After the first city is chosen (call this c_1), the edge of greatest distance between it and its two neighbours, is removed. Refer to the affected city as c_2. c_1 is then reconnected to a city, c_3, that is closest to it. However, in so doing, a sub-tour is created, therefore one of the edges from c_3 is uniquely removed. Denote the other city to this edge as c_4. c_4 is then reconnected to c_2 in order to generate a valid Hamiltonian circuit. It is evident that all that occurs is a 2-opt between c_2 and c_3. As these authors only deal with the symmetric version of the problem, very few links are effectively broken. However, if applied to the ATSP the change from one solution to another could be very large. Using relatively small test instances, their results showed plenty of room for potential improvement to the technique.

Chen, Zhu, Yang and Lu [8] build on the work of Chen, Lu and Chen [7] who solve the standard TSP, by applying EO to the ATSP. Their "Improved Extremal Optimisation" (IEO) consists of two main steps; extremal dynamics (i.e., the EO algorithm) and co-operative optimisation. The latter, it is claimed, leads to the named improvement. The authors assert that for problems having encoding constraints (such as a permutation for travelling salesmen problems), by making a local transition, the state of other decision variables are necessarily affected. For example, for a permutation problem, a 1-opt transition will affect the positions of proceeding values in the structure. In this case, "co-operative optimisation" can be seen as more of a necessity to ensure feasible solutions. Furthermore, the authors map the ATSP onto a multi-entity physical system. This largely affects their ranking strategy, which is discussed in Section 3.2 of this paper. The local search operator used in their paper is 3-opt. At each iteration, a "bad" city with high energy is chosen. From this, the operator considers all possible 3-opt transitions from this point alone. If the best of these produces a new solution that is the best found to date, the transition is performed.

---

[1]This assumes that the problem is separable, and as such, the cost of each element can be readily calculated.

Should that not be the case, with a certain probability, a random transition from the neighbourhood is performed. Other than this, a greedy/descent search strategy, that locates the local optimum, is used instead.

3   Neighbours, Rankings and Transitions

As previously mentioned, EO's mechanics do not lend themselves easily to permutation problems, such as the ATSP. By augmenting the canonical algorithm with some generalisable heuristics, this can be alleviated. This section therefore outlines techniques for managing the use, and taking advantage of, selecting local search operators; generating primary and secondary endpoints; and local search refinement.

3.1   Local Transition Operators and their Selection

At each iteration of the search, canonical EO will choose a probabilistically weak solution component, and change its value to a random value. This mirrors certain aspects of nature where non-adaptive or unfit species are replaced, within an environment, by new species [1]. For highly structured problems, such as those requiring permutations as solutions, the canonical strategy requires modification. For permutation problems, at least two "endpoints" are required so that the resultant solution is feasible. For example, swapping the positions of two cities in the travelling salesman problem ensures this. At the very least, a single point, chosen by EO's mechanics is required. This is referred to as the *primary endpoint*, while the other point(s) is/are the *secondary endpoint(s)*[2].

Given that a primary and a secondary endpoint(s) have been chosen (as discussed in the next two subsections), there are a variety of local transition operators that may be applied to permute solutions. The work in this study is based on the use of the following four operators. While they are described in terms of the application problem, they are each appropriate for other permutation problems.

- *1-opt* – A city is moved from its current position in the tour to another. For the ATSP, only three edges are removed and three other edges are added in this operation.

- *2-opt* – A 2-opt removes two edges. Each edge is replaced by a new one that connects to the node of the other removed edge. There is a unique way of doing this without creating two sub-tours. In essence, 2-opt corresponds to an inversion operation. As potentially a large number of links are inverted, this type of operation can lead to a large overall change for the ATSP, whereas, it will be relatively small for the symmetric TSP.

- *Swap* – Two cities in the tour change position. This breaks/removes four existing links and creates four replacements.

- *3-opt* – Three edges are removed, and are replaced by another three edges. If no intermediate edges are to be inverted, there is a unique way to do this. Unlike the 2-opt, however, no intermediate edges are inverted.

The question becomes how to select a transition operator at each iteration. While it is possible to do this any number of ways, one way that is commonly used for stochastic choices in meta-heuristic algorithms is roulette wheel selection. Selecting the associated probabilities of the transition operators becomes part of the experimental framework and is hence described in the next section.

3.2   Primary Endpoint Selection

To select the primary endpoint at each iteration, one of the key aspects of the EO algorithm, ranking of the solution components, is used. An intuitive, and perhaps somewhat naïve approach to ordering the set

---

[2]If an operation such as 3-opt is used, there will be more than one secondary endpoint.

of cities is to simply consider the distance metric. However, this does not take into consideration relative differences in edge size and an individual city's need to be connected to its nearest neighbour. Therefore both Boettcher and Percus [3, 4] and Chen et al. [8] propose measures, albeit in slightly different ways, to address these issues. Boettcher and Percus' "frustration measure" ranks cities on the degree to which they are separated from their two ideal neighbour cities. The greater the number of links in the sequence/tour that a city is removed from its ideal neighbours, the more likely it is to be selected to change by EO. The ranking is based on the frustration measure given in Equation 1.

$$\lambda_i = \frac{3}{p_i + q_i} \tag{1}$$

Where: λ_i is the frustration measure of the ith city in the sequence, p_i is the ranked order (in terms of the list of closest cities to the city in the ith position) of the proceeding city in the tour sequence and q_i is similar to p_i except that it is for the city that precedes i.

Ideally, $\lambda_i = 1, \forall i$ which means that every city is connected to its closest city. However, optimal TSP solutions will rarely obey this property.

Chen at al. [8] however, model the TSP as a physical system in which the energy ground state corresponds to an idealised solution in which each city is connected to its closest neighbour. This is given by Equation 2.

$$e_i = p_i - \min_{j \neq i}(d_{ij}) \tag{2}$$

Where: e_i is the energy associated with the ith city in the tour, p_i is the length of the forward edge from the ith city in the tour and d_{ij} is the distance between city i^{th} and j^{th} cities in the tour.

Both of these techniques are implemented and tested in the next section of the paper.

3.3 Secondary Endpoint Selection

The previous subsection described how the primary endpoint is chosen using the canonical EO selection mechanism. There are two distinct ways in which the secondary endpoint or endpoints (in the case of 3-opt) can be selected. For ease of discussion, the singular "endpoint" will be used even when referring to 3-opt's two endpoints.

The first way is very simple. A random city is chosen as the secondary endpoint. Of course this needs to be different from the primary endpoint, otherwise no transition will take place.

The second way is more complex and is referred to as the *neighbourhood* approach. Rather than choosing another city/location at random, each city/location is evaluated and the best one is chosen. The procedure outlined in Algorithm 2 is employed and is applicable to all four operators.

Algorithm 2 The generic algorithm for performing a neighbourhood search. This assumes that the objective is to be minimised.

Set *best_cost* to a large value
for all possible neighbours of this operator using the primary endpoint **do**
    Perform the neighbour operation and record the change to the objective function as *cost*
    if *cost* < *best_cost* **then**
        Record the secondary endpoint(s) of this operation
        best_cost = *cost*
    end if
    Restore the original solution by reversing the operator
end for
return the secondary endpoint(s) of the best neighbourhood operation and its cost
end

Name	N	Best Known Cost
ftv55	56	1608
ftv64	65	1839
ft70	71	38673
kro124p	100	36230
ftv170	171	2755
rbg323	323	1326
rbg358	358	1163
rbg403	403	2465
rbg443	443	2720

Table 1: The ATSP instances used in this work.

In this paper, operators that use neighbourhoods are designated using the prefix n. For example, $n1$-opt is the neighbourhood version of 1-opt.

3.4 Local Search

Like all meta-heuristics, EO requires a subordinate refinement algorithm to move its solution to local optimality. Consistent with other works around the travelling salesman problem (such as Iorache [10], Merz and Freisleben [11] and Stützle, Grün, Linke and Rüttger [16]), 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).

4 Computational Experiments, Results and Analysis

The purpose of this section is twofold. As there are effectively eight local search operators (single and neighbourhood) with associated probabilities and a choice of two ranking methods, a large number of parameter combinations are possible. Section 4.2 presents a hierarchical parameter tuning regime that is designed to examine the effects of various parameter choices to produce good parameter combinations. Section 4.3 gives the results of the final parameter sets on the same test suite of problems as used by Chen et al [8].

4.1 Test Environment

Table 1 lists the characteristics of the ATSP instances used for these experiments. Each problem instance will be run across ten random seeds to give statistically valid results. A run is terminated if it reaches the best known solution cost for a problem instance or a maximum number of iterations has been reached. Results are reported as relative percentage deviations (RPDs) from the best known solution cost for each problem instance. Formally this is given $\frac{a-b}{b} \times 100\%$ where a is the obtained cost and b is the best known cost. A value of 1.4 is used for EO's τ parameter, as it has been found to be a good value for similar problems [14, 15].

4.2 Hierarchical Parameter Tuning

Having a large number of parameters, some of which have an infinite number of settings, can result in an extensive and unproductive search of its own. Hence, an iterative, hierarchical scheme is used to systematically look at the effects of various parameter settings.

A subset of the problem instances, namely ftv55, ft70 and kro124p will be used for this exploratory phase. This approach of using a subset of the test problems to determine suitable parameter values is due to the potentially large amount of time taken by parameter tuning [2]. A relatively small number of

Problem			1-opt	swap	2-opt	3-opt	$n1$-opt	$n$swap	$n2$-opt	$n3$-opt	
ftv55	C	Min	0	0.31	20.4	1.93	4.48	4.66	1.93	1.93
		Med	0	7.03	24.75	15.11	7.65	8.02	10.82	13.9
		Max	3.92	11.82	29.42	21.08	11.32	13.12	39.68	20.52
	B	Min	0	3.17	18.41	10.07	1.55	5.22	2.18	3.36
		Med	0	8.96	24.41	16.17	7.8	6.69	7.87	14.61
		Max	1.31	10.32	27.86	22.57	12.31	11.82	18.1	20.27
ft70	C	Min	0.44	3.68	12.1	1.01	1.77	1.11	1.96	1.63
		Med	0.88	4.59	13.15	2.97	2.14	2.98	3.51	3.22
		Max	1.08	4.96	14.68	5.79	4.18	3.68	9.18	6.22
	B	Min	0.6	3.35	11.32	1.91	1.43	2.06	2.17	2.45
		Med	0.84	4.26	13.71	4.45	2.16	2.69	3.24	4.18
		Max	1.13	5.08	14.8	5.56	3.29	6.24	10.21	5.15
kro124p	C	Min	0.55	16.45	14.46	6	3.11	3.16	3.02	5.79
		Med	2.91	21.15	18.15	13.22	6.92	8.67	9.36	13.57
		Max	5.79	22.45	19.93	17.8	11.36	11.47	15.05	17.8
	B	Min	0.18	16.2	12.98	8.49	5.35	5.06	5.14	8.5
		Med	3.48	20.87	18.2	11.62	6.11	8.75	9.36	12.61
		Max	5.8	23.09	21.25	19.48	10.75	14.26	15.14	18.39

Table 2: Results for ftv55, ft70 and kro124p. Note that "Min", "Med" and "Max" refer to minimum, median and maximum respectively. Results are recorded as RPDs.

iterations are used to gauge the performance of each setting. In this case, the number of iterations is set to 10000. In the first instance, the effects of the following need to be ascertained:

1. The effect of each of the eight transition operators used in isolation.

2. The effect of each ranking scheme.

Table 2 shows the results collectively for the three problem instances. Each table has all four combinations of ranking scheme and endpoint type for each type of operator. Note that 'C' refers to Chen et al. [8] ranking and 'B' for Boettcher and Percus [3] ranking. As the data are non-normally distributed, the Kruskal-Wallis technique is used to detect if there are any significant differences detected because of any of these attributes. At the $\alpha = 0.05$ level the following was observed:

- No significant differences occurred the ranking method. In terms of the Kruskal-Wallis ranks, the C setting produced the overall lowest RPD values.

- A significant difference was present due to the operator. Using the Kruskal-Wallis ranks revealed the following order: 1-opt $\prec$ $n1$-opt $\prec$ $n$swap $\prec$ $n2$-opt $\prec$ 3-opt $\prec$ swap $\prec$ $n3$-opt $\prec$ 2-opt.

Given that an order of goodness has been derived for the eight transition operators, it is now possible to try various combinations of these operators to determine if better results can be obtained than those given in Table 2. There are actually two questions here:

1. Do combinations of good operators give good solutions, and conversely, do combinations of poor operators give poor solutions?

2. Does the mere use of more than just one operator help to improve solution quality?

To answer these questions, the following scheme is devised. Three groups of three operators, labelled, "best", "middle" and "worst" are used with various probability settings. The groups are (1-opt, $n1$-opt, $n2$-opt), ($n$swap, $n2$-opt, 3-opt) and (swap, 3-opt, 2-opt) respectively. The probability settings are $(0.8, 0.1, 0.1)$, $(0.6, 0.2, 0.2)$, $(0.5, 0.25, 0.25)$ and $(0.4, 0.3, 0.3)$.

Problem	Metric	Transition Set			
		(0.8,0.1,0.1)	(0.6,0.2,0.2)	(0.5,0.25,0.25)	(0.4,0.3,0.3)
ftv55	Min	0	0	0	0
	Med	0	0	0	0
	Max	0	0	0	0
ft70	Min	0.31	0.01	0.02	0.02
	Med	0.47	0.36	0.12	0.23
	Max	0.54	0.49	0.32	0.35
kro124p	Min	0.03	0.03	0.03	0.03
	Med	0.5	0.09	0.99	0.03
	Max	3.5	2.22	2.91	3.27

Table 3: Results for the best group (1-opt, $n1$-opt, $n2$-opt).

Problem	Metric	Transition Set			
		(0.8,0.1,0.1)	(0.6,0.2,0.2)	(0.5,0.25,0.25)	(0.4,0.3,0.3)
ftv55	Min	1.24	1.18	3.48	4.54
	Med	5.66	6.59	6.06	8.33
	Max	17.16	11.13	18.1	14.37
ft70	Min	0.74	0.63	0.74	0.48
	Med	1.68	2.33	1.65	2.81
	Max	3.15	3.82	4.25	5.27
kro124p	Min	2.29	4.39	5.74	3.89
	Med	8.07	7.47	7.55	5.62
	Max	10.5	17.17	9.57	10.56

Table 4: Results for the middle group ($n$swap, $n2$-opt, 3-opt).

From the results presented in Tables 3, 4 and 5, the following statements can be made. If good operators are combined, better results are achieved, whereas, if poor operators are used in conjunction, typically even worse results are produced. Therefore, question one is answered in the affirmative for the problem instances tested here.

Of the most interest is the best group, as it is able to help to answer the second question. Applying Kruskal-Wallis reveals that the transition set $(0.4, 0.3, 0.3)$ outperforms the others. This is an interesting finding as it demonstrates a mixture of operators, even with a de-emphasis on the best operator, is good for this problem. To investigate this further, two more sets were applied. These are $(0.34, 0.33, 0.33)$ and $(0.2, 0.4, 0.4)$. These results are given in Table 6. Both of these were slightly better on some instances than transition set $(0.4, 0.3, 0.3)$. As a final answer to question two, for this problem and these problem instances at least, the answer appears to be yes. This will be tested even more in the next group of

Problem	Metric	Transition Set			
		(0.8,0.1,0.1)	(0.6,0.2,0.2)	(0.5,0.25,0.25)	(0.4,0.3,0.3)
ftv55	Min	8.15	2.86	8.21	9.08
	Med	13.03	14.27	12.19	12.66
	Max	16.73	15.3	15.05	15.24
ft70	Min	4.81	5.57	6.23	4.44
	Med	6.64	7.36	7.46	6.99
	Max	8.09	8.69	8.21	7.97
kro124p	Min	17.37	15.82	11.12	9.53
	Med	20.06	18.08	16.39	14.19
	Max	25.07	19.69	18.62	17.96

Table 5: Results for the worst group (swap, 3-opt, 2-opt).

Problem	Metric	Transition Set	
		(0.34,0.33,0.33)	(0.2,0.4,0.4)
ftv55	Min	0	0
	Med	0	0
	Max	0	0.62
ft70	Min	0	0
	Med	0.2	0.18
	Max	0.3	0.28
kro124p	Min	0.03	0
	Med	0.03	2.08
	Max	3.55	2.63

Table 6: Extended results for the best group.

experiments.

4.3 Final Results

From the preliminary results, it is evident, for this problem at least, that the use of multiple transition operators is effective. The aim of this part is to determine the effect of the three best transition sets on the entire ATSP test suite, with an extended number of iterations (50000). Additionally, for purposes of comparison, one other test will be performed. This is to use pure 1-opt as it was identified to be the best amongst the single use operators (denoted as (1,0,0)). This will serve as a control. Table 7 presents the entire set of results.

In terms of the four variants in Table 7, it is noted that the three that use the combination of transition operators are able to produce the best known solution for each problem instance. The variant that uses 1-opt alone does so for 4/10 instances. Given the increased number of available iterations (over the initial 10000) confirms that EO is capable of continual exploration and does not get trapped in local optima. A Kruskal-Wallis analysis reveals that a significant difference in performance does exist between the four variants, with the 1-opt variant receiving the highest (worst) rank. The order becomes $(0.34,0.33,0.33) \prec (0.4,0.3,0.3) \prec (0.2,0.4,0.4) \prec (1,0,0)$. This suggests for this problem and these instances, combinations of local search transition operators provides a performance advantage. The instances for which the best known solution is consistently found (such as rbg443), typically do so within a few hundred iterations. This consumes only a few seconds of computational time on a standard PC.

In comparison to the work of Chen at al. [8] it can be seen that both sets of approaches are able to generate the best known solution to these test problem instances. In four of its instances, it is more consistent in receiving the best known solution, while the solver presented here is more consistent than it on rbg403. It would be interesting to combine aspects of their work, particularly their "cooperative optimization" (specifically step 3 of their algorithm in Section 3.2, p. 4462) with the multiple neighbourhood concept as outlined here.

5 Conclusions

The choice of local search transition operators plays a critical part of the optimisation process for meta-heuristic algorithms. In many works, a single operator is used, its choice often being determined by the state of the literature, while other parts of the meta-heuristic search algorithm are explored or modified. In contrast, this paper has made an examination of operators, and importantly, the use of multiple operators for a difficult combinatorial problem, the asymmetric travelling salesman problem. The interesting finding was that a combination of local transition operators produced better overall results, than a single good operator. This was achieved by using a simple probabilistic selection method with probabilities derived by a small set of investigations. In addition to this, it was shown that the choice of transition

		(0.34,0.33,0.33)	(0.4,0.3,0.3)	(0.2,0.4,0.4)	(1,0,0)
ftv55	Min	0	0	0	0
	Med	0	0	0	0
	Max	0	0	0	0
ftv64	Min	0	0	0	0
	Med	0.49	0.16	0.54	0.54
	Max	0.6	0.6	0.6	1.03
ft70	Min	0	0	0	0.14
	Med	0	0.01	0.01	0.49
	Max	0.15	0.2	0.03	0.82
ftv70	Min	0	0	0	0.36
	Med	0.36	0.46	0.51	0.56
	Max	0.56	0.46	0.56	0.92
kro124p	Min	0	0	0	0.14
	Med	0	0.02	0	1.38
	Max	0.34	1.92	0.03	3.82
ftv170	Min	0	0.22	0	1.52
	Med	0.91	0.78	0.58	4.01
	Max	4.32	4.14	5.05	6.72
rbg323	Min	0	0	0	0.36
	Med	0	0	0	0.56
	Max	0	0	0	0.92
rbg358	Min	0	0	0	0
	Med	0	0	0	0
	Max	0	0	0	0
rbg403	Min	0	0	0	0
	Med	0	0	0	0
	Max	0	0	0	0
rbg443	Min	0	0	0	0
	Med	0	0	0	0
	Max	0	0	0	0

Table 7: The final results for the selected variants using 1-opt, $n1$-opt, $n2$-opt.

operators led to greater improvements in locating good solutions than component ranking schemes or how operator endpoints were generated.

There are a number of directions in which this work leads. Firstly, a number of different combinatorial problems need to be re-examined in terms of their local search operator choice. It may very well be the case that the introduction of multiple neighbourhoods may lead to increased performances and a deeper understanding of why certain operators work well with particular problems. One important aspect that needs examination is the transition selection mechanism. In this paper, a simple probabilistic model was used. However, more sophisticated choices based on the concepts of landscape fitness analysis [9] and self-adaptation [13] need to be investigated. Also, as identified in the previous section, a hybridisation of the Chen et al. [8] cooperative optimisation and multiple neighbourhoods may lead to further performance improvements for ATSP, and possibly a greater range of problems as well. Finally, to progress the approach outlined in this paper to being a full memetic algorithm requires a) the examination of local search heuristics to achieve fine grain search, and b) the implementation of an EO population model (such as Chen et al. [6] and Randall et al. [15]).

References

[1] P. Bak and K. Sneppen. Punctuated equilibrium and criticality in a simple model of evolution. *Physical Review Letters*, 71:4083–4086, 1993.

[2] M. Birattari, M. Chiarandini, M. Saerens, and T. Stutzle. Learning graphical models for parameter tuning. Technical Report TR/IRIDIA/2011-002, Institut de Researches Interdisciplinaires et Development en Intlligence Artificelle, Bruxelles, Belgium, 2011.

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

[4] S. Boettcher and A. Percus. Nature's way of optimizing. *Artificial Intelligence*, 119:275 – 286, 2000.

[5] E. Burke, R. Cowling, and R. Keuthren. Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In *Ant Colony Optimization and Swarm Intelligence*, volume 2037 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2001.

[6] M. Chen, Y. Lu, and G. Yang. Population-based extremal optimization with adaptive lvy mutation for constrained optimization. In Y. Wang, Y. Cheung, and H. Liu, editors, *Computational Intelligence and Security*, volume 4456 of *Lecture Notes in Computer Science*, pages 144–155. Springer Berlin / Heidelberg, 2007.

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

[8] Y. Chen, Y. Zhu, and Y. Lu. Improved extremal optimization for the asymmetric traveling salesman problem. *Physica A:Statistical Mechanics and its Applications*, 390:4459–4465, 2011.

[9] J. He, C. Reeves, C. Witt, and X. Yao. A note on problem difficulty measures in black-box optimization: Classification, realizations and predictability. *Evolutionary Computation*, 15(4):435–443, 2007.

[10] S. 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*, PPSN'10, pages 81–90, Berlin, Heidelberg, 2010. Springer-Verlag.

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

[12] P. Moscato. Memetic algorithms: A short introduction. In D. Corne, M. Dorigo, and F. Glover, editors, *New ideas in optimization*, pages 219–234. McGraw-Hill, Maidenhead, UK, England, 1999.

[13] A. Qin and P. Sugnathan. Self-adaptive differential evolution algorithm for numerical optimization. In *Proceedings of the Congress on Evolutionary Computation*, pages 1781–1791, 2005.

[14] M. Randall. Enhancements to extremal optimisation for generalised assignment. In M. Randall, H. Abbass, and J. Wiles, editors, *The Third Australian Conference on Artificial Life*, volume 4828 of *Lecture Notes in Artificial Intelligence*, pages 369–380, Berlin, 2007. Springer.

[15] M. Randall, T. Hendtlass, and A. Lewis. Extremal optimisation for assignment type problems. In A. Lewis, S. Mostaghim, and M. Randall, editors, *Biologically-inspired Optimisation Methods: Parallel Algorithms, Systems and Applications*, volume 210 of *Studies in Computational Intelligence*, pages 139–164. Springer-Verlag, 2009.

[16] 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*, PPSN VI, pages 661–670, London, UK, 2000. Springer-Verlag.