New Order-Based Crossovers for the Graph Coloring Problem

Christine L. Mumford

Cardiff University, School of Computer Science
C.L.Mumford@cs.cardiff.ac.uk
http://www.cs.cardiff.ac.uk/

Abstract. Huge color class redundancy makes the graph coloring problem (GCP) very challenging for genetic algorithms (GAs), and designing effective crossover operators is notoriously difficult. Thus, despite the predominance of population based methods, crossover plays a very minor role in most state-of-the-art approaches to solving the GCP. Two main encoding methods have been adopted for heuristic and GA methods: direct encoding, and order based encoding. Although more success has been achieved with direct approaches, algorithms using an order based representation have one powerful advantage: every chromosome decodes as a feasible solution. This paper introduces some new order based crossover variations and demonstrates that they are much more effective on the GCP than other order based crossovers taken from the literature.

1 Introduction

The graph coloring problem (GCP) is a well studied combinatorial optimization problem. It involves finding a minimum set of colors for the vertices of a given graph, so that no two adjacent vertices have the same color. Interest in the GCP is broadly based and the field is highly competitive. In 1993 the GCP was the subject of a Discrete Mathematics and Theoretical Computer Science (DIMACS) implementation challenge, [11], in which the best algorithms of the day were pitted against each other on a collection of large and very difficult benchmark instances. Since 1993 enthusiasm for the GCP has not diminished, and new approaches continue to be developed. As an archetypal set partitioning problem, the GCP provides a useful testbed for techniques applicable more widely to real world problems such as timetabling [2] and frequency assignment [16].

The GCP is NP-hard, thus heuristic and metaheuristic methods are appropriate techniques for solving the problem. Two main encoding methods can be identified: direct encoding, and order based encoding. With direct encoding arbitrary colors are assigned and heuristics used to recolor the vertices in an attempt to improve the solution. On the other hand, order based approaches organize vertices as permutation lists, and rely on a greedy decoder to assign the colors in a methodical way. Although algorithms using a direct approach have enjoyed more success recently, order based techniques have one powerful advantage: every permutation decodes as a feasible solution. Direct approaches work to minimize and eventually eliminate conflicts but do not guarantee legal solutions.

The main contribution of the present paper is to introduce two new and effective order based crossover/local search combinations: *Merge Independent Sets (MIS)* and *Permutation One Point (POP).* The success of the new operators is demonstrated by comparing their performance on DIMACS benchmarks with that of well known order based crossovers for the GCP taken from the literature.

2 Summary of Related Work

As mentioned above, the most successful current approaches to solving the GCP use a direct representation with conflict minimization as the goal: i.e., given k colors, a coloring is sought which minimizes the number of adjacent vertices bearing the same color. Most commonly, a genetic (or population based) algorithm (GA) is used in combination with some form of local search. However, because graph coloring is essentially a set partitioning problem with arbitrary color labels assigned to the individual sets, crossover has proven a huge challenge. Indeed, several population based approaches do not incorporate crossover at all, [9, 14]. For others, crossover plays only a limited role, [3, 7]. Exceptional among recent conflict minimization techniques is the hybrid coloring algorithm of Galinier and Hao [8]. In this case a novel crossover operator makes a significant contribution to the success of the approach. Furthermore, competitive results have been reported on large DIMACS benchmarks.

Although rather less successful than direct encoding methods, order based techniques for the GCP have a very long history. These methods rely on a simple greedy algorithm to transform a permutation of vertices into a legal coloring, and it is thus the role of good ordering (or reordering) heuristics to present the greedy algorithm with a suitable permutation that it can transform into a high quality solution. The simplest and fastest such techniques generate an ordering in one go, usually by ensuring that the more heavily constrained vertices are placed before those that are less constrained [13, 17]. A somewhat more sophisticated technique, known as *DSatur*, [5] operates in two stages. The first stage produces a list of vertices sorted by decreasing degree; and the second stage selects vertices from this list according to their *saturation*, i.e., the number of distinct colors already assigned to adjacent vertices. Unfortunately, despite their attractiveness in terms of speed and simplicity, none of the above mentioned simple ordering techniques produces very good results on large benchmarks.

Rather more successful than "one go" ordering heuristics based on vertex degree, are the iterative reordering heuristics of Culberson and Luo, [4]. These methods do not rely on vertex degrees or saturation. Instead, beginning with an arbitrary permutation and greedy allocation of colors, Culberson and Luo's heuristics operate by grouping and rearranging color classes along the permutation list. Of particular significance is a rare property possessed by each of Culberson and Luo's reordering heuristics for the GCP: it is impossible to get a worse coloring by rearranging the color classes, and it is possible that a better coloring (using fewer colors) may result. Capitalizing on this property, the authors applied a random mix of various reordering heuristics repeatedly to individual problems, and watched the solutions gradually improve. They called their algorithm *iterated greedy*. Disappointingly, however, in spite of its elegance, the iterated greedy technique achieves only moderate success on large graph benchmarks.

We now move on to consider order based GAs for the GCP. In one of the earliest and best known studies of this type Davis developed new order based crossover and mutation operators especially for the GCP, [12]. However Davis' work predates the DIMAC challenge and he used a very specialized type of graph (with weighted edges) to test his algorithms. Realizing difficulties in designing effective crossover operators for the GCP, Eiben *et al*, [6] developed an order based evolutionary algorithm with mutation only. Once more, though, the approach was tested only on a very specialized type of graph coloring problem: the three color problem. More recently, Anderson and Ashlock [1] have introduced a crossover called "merging crossover" (MOX) which shows some promise for the GCP, although, once again, the authors did not present results for literature benchmarks. The present author is heavily indebted to many of the above mentioned researchers for their insight, and many of their ideas have been incorporated into the present work.

3 The GA Framework

A simple steady-state GA is used as a framework for our comparative study, which concentrates only on crossover operations. There are few parameters to set using this approach. For example, no global fitness function is used and crossover occurs at 100 % with no mutation. At the start of the procedure a population of N random permutations is generated. If the GCP instance has n vertices, then each chromosome will consist of a permuted list of the integers $\{1, 2, 3, \ldots n\}$. Once the initial population is created, the individual members have to be evaluated, according to the performance measure described later. Within the main generation loop, each member of the population is selected in turn and paired in crossover with a second individual selected (uniformly) at random. The performance measure of the resulting single offspring is then compared with that of its weaker parent. If the new offspring is better than its weaker parent it replaces it in the population, otherwise it dies. The GA is run for a fixed number of generations, where a generation is defined as N trials of crossover, one led by each member of the population in turn.

The number of colors (chromatic number) is probably not the best measure of progress to use, given that many colorings will produce identical values. In this paper we will adopt the progress measure used by Culberson and Luo [4]: $\sum_1^n c_i + nc$. In this equation the *coloring sum* (i.e., $\sum_1^n c_i$, where c is the color assigned to vertex i) is added to the term nc, where n is the number of vertices and c the number of colors. The main idea is encourage large color classes to grow even larger at the expense of the smaller classes, in the hope that eventually some classes will lose their remaining vertices and disappear altogether. We shall see later that the two operations taken from Culberson and Luo, [4], "sort independent sets" followed by "largest first", ensure that the color sets are presented in an optimum sequence for minimizing the coloring sum. The term nc is added to ensure that improved colorings are always reflected in the measure of progress.

3.1 The Representation and Greedy Decoder

Chromosomes consist of permutations of the n vertices, and the greedy decoder colors each vertex in sequence, using the first available color from an ordered set (i.e., each color is identified by an integer label, $0, 1, 2, 3, \ldots$). Figure 1 a) and 1 b) illustrate this process using a small graph with 12 vertices and 14 edges, Figure 1 b) giving a typical random permutation of the vertices from Figure 1 a) and also the resulting greedy coloring. Note: an efficient version of the greedy algorithm has been implemented using linked lists to keep track of vertices already assigned to color classes, as advised in [4]. The remaining parts of Figure 1 illustrate the stages of the optional local search procedure described below.

3.2 The Local Search

The local search uses Culberson and Luo's [4] "largest first" and "sort independent set" heuristics. Its purpose in the present study is twofold: firstly to reduce the value of the performance measure, and secondly to improve the correlation of the chromosomes in the population. Figure 1 c) shows the permutation list sorted in non-descending sequence of color label, and 1 d) gives the position following the application of the "largest first" heuristic: i.e., the list is rearranged in non-ascending sequence of color class size. Following the advice given in [4], the sequence of color classes of identical size is reversed. Note that the application of "largest first" will normally reduce the value of the performance measure. In Figure 1 f) vertices are randomly "shuffled" within (but not between) independent sets. Finally, the greedy algorithm is applied to the new arrangement in f) and the result is shown in 1 g). Interestingly, vertices 4 and 1 are reassigned lower color labels, further reducing the magnitude of the performance measure. In the present study the local search loop is iterated 5 times.

3.3 The Crossover Operators used for the Comparative Study

A genetic algorithm with a population of permutation lists requires a crossover technique that preserves building blocks [10] appropriate for the GCP. Of particular relevance for the greedy decoder, is that some items precede others in the permutation list. Historically the operators *cycle crossover (CX)* [15] and *uniform order based crossover (UOBX)* [12] seem worthy of consideration. CX is good at preserving absolute positions of vertices, and every vertex in the offspring list will occur in exactly the same position in one or other of its parents.

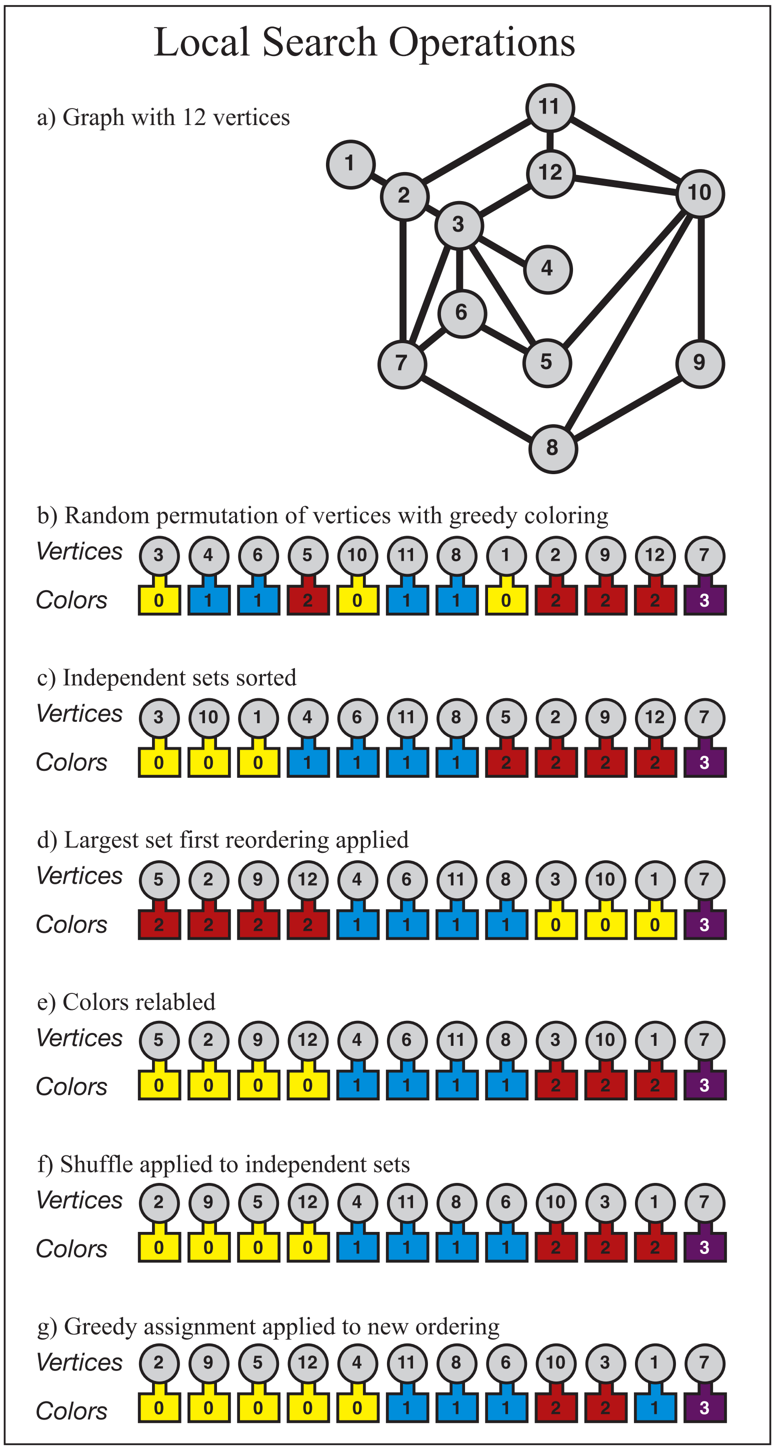
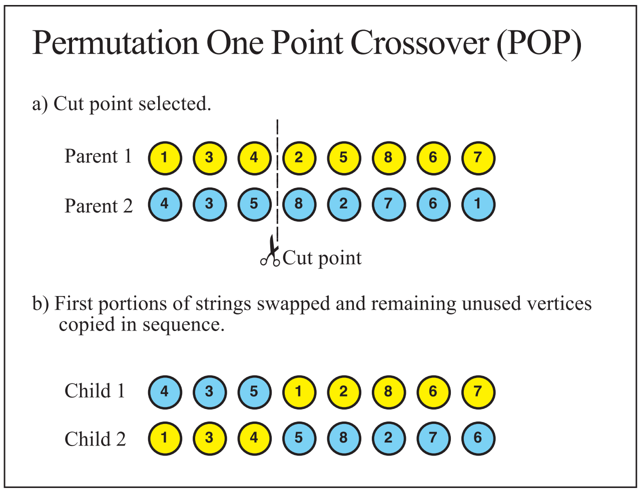

Local Search Operations
a) Graph with 12 vertices
1 2 3 4 5 6 7 8 9 10 11 12
b) Random permutation of vertices with greedy coloring
Vertices 3 4 6 5 10 11 8 1 2 9 12 7
Colors 0 1 1 2 0 1 1 0 2 2 2 3
c) Independent sets sorted
Vertices 3 10 1 4 6 11 8 5 2 9 12 7
Colors 0 0 0 1 1 1 1 2 2 2 2 3
d) Largest set first reordering applied
Vertices 5 2 9 12 4 6 11 8 3 10 1 7
Colors 2 2 2 2 1 1 1 1 0 0 0 3
e) Colors relabled
Vertices 5 2 9 12 4 6 11 8 3 10 1 7
Colors 0 0 0 0 1 1 1 1 2 2 2 3
f) Shuffle applied to independent sets
Vertices 2 9 5 12 4 11 8 6 10 3 1 7
Colors 0 0 0 0 1 1 1 1 2 2 2 3
g) Greedy assignment applied to new ordering
Vertices 2 9 5 12 4 11 8 6 10 3 1 7
Colors 0 0 0 0 0 1 1 1 2 2 1 3

Fig. 1. Various operations by Culberson and Luo, [4], used in the local search procedure

CX has proven effective in the related frequency assignment problem [16]. UOBX was developed by Davis with the GCP in mind and is good at preserving relative positions and orderings.



Permutation One Point Crossover (POP)
a) Cut point selected.
Parent 1
1 3 4 2 5 8 6 7
Parent 2
4 3 5 8 2 7 6 1
Cut point
b) First portions of strings swapped and remaining unused vertices copied in sequence.
Child 1
4 3 5 1 2 8 6 7
Child 2
1 3 4 5 8 2 7 6

Fig. 2. POP Crossover

Some new variations of the well known *order crossover (OX)* [15] are also tried here. The basic idea is taken from the simple one point crossover of the "standard" bit string GA, which simply selects two parents and a cut point. The first portion of parent 1 up to the cut point becomes the first portion of offspring 2, in the normal way. However, the remainder of offspring 2 is obtained by copying the vertices absent from the first portion of the offspring in the same sequence as they occur in parent 2 (see Figure 2). We will call this crossover *permutation order based crossover (POP)*. Furthermore, we will identify two variants, POP1 and POP2, which differ slightly in the way the cut point is selected: for POP1 it is chosen at random and can appear anywhere in the list, but for POP2 the cut point is restricted to a boundary between two color classes. Of course the application of POP2 is dependent on having previously sorted the color classes.

Merging crossover (MOX) was presented by Anderson and Ashlock, [1]. Initially two n element parents are randomly merged into a single $2n$ element list. The first occurrence of each value in the merged list gives the ordering of elements in the first child, and the second occurrence in the second child. MOX is illustrated in Figure 3. Anderson and Ashlock point out the following property of MOX: if and element, a precedes another element b in both parents, then it follows that a will precede b in both children.

Merging independent sets (MIS) is a new crossover, adapted from MOX. It requires that the color sets are first grouped together in both of the parents, as illustrated in Figure 1 c). MIS then proceeds in the same way as MOX, but whole color sets are copied from the parents to the merged list in one go, rather than individual vertices. The merged list is split in exactly the same way as for

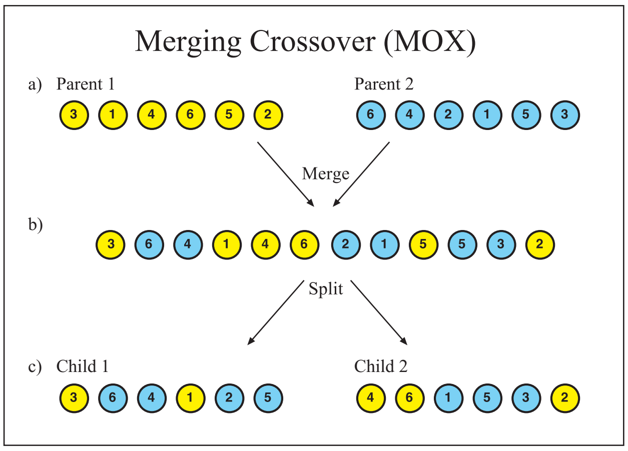


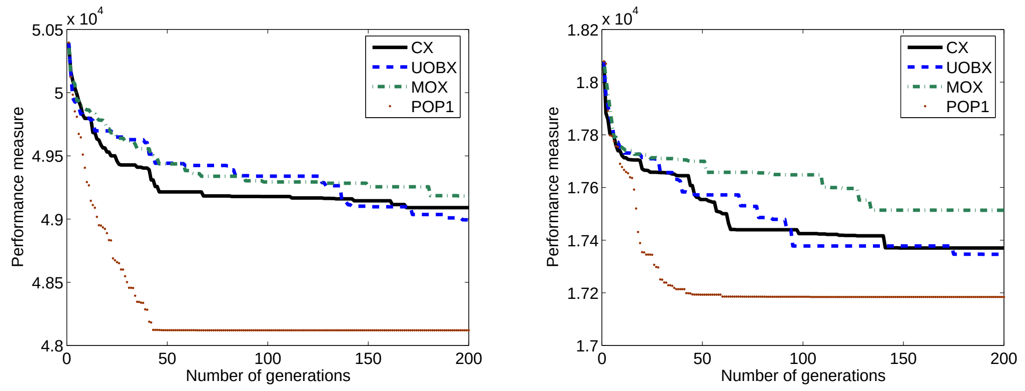
Fig. 3. MOX Crossover by Anderson and Ashlock, [1], used as a basis for the new MIS crossover

MOX, with the first occurrence of each vertex appearing in the first offspring and the second occurrence in the second offspring. The idea of MIS is to better preserve the parents' color classes than MOX.

4 Results

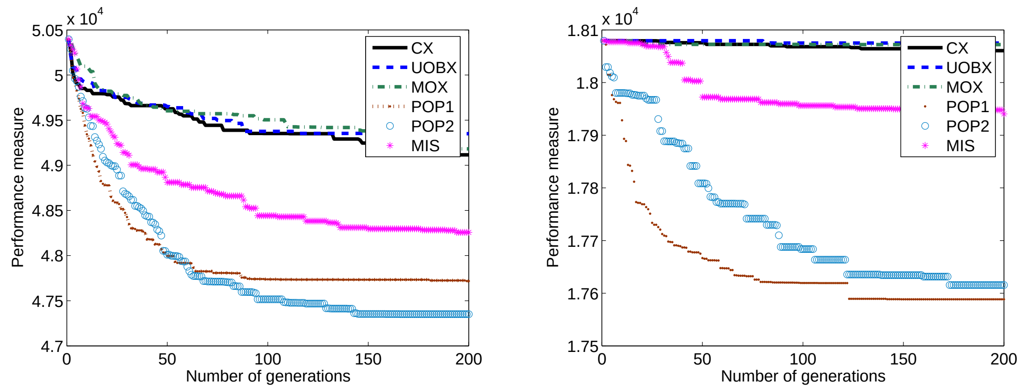
Three sets of experiments were conducted to assess the viability of the various crossover operators for the GCP on two benchmarks from the literature, DSJC500.5 and le450_15c. The first set of experiments used a basic order based approach without incorporating the local search. Color classes were neither collected nor sorted. Only CX, UOBX, MIS and POP1 could be compared here, because the other operators rely on sorted color classes. Local search did not form part of the second set of experiments either, although the color classes were sorted and grouped to make it possible to test all the crossovers in our study. Finally, the third set of experiments included the full local search. Results for all the experiments are displayed graphically as best-so-far curves averaged over 10 replicate runs, see Figures 4, 5 and 6. For each run a population of 250 was used and the GA run for 200 generations. Clearly the best results are obtained when local search is used in Figure 6, with MIS a clear winner on DSJC500.5 and POP1 on le450_15c.

To complete the study, a final set of experiments were performed to indicate the potential of the new techniques on seven large DIMACS benchmarks. That MIS and POP1 perform well compared to the other order based crossover operators has already been established, but the results could surely be improved with longer runs and the introduction of a mutation operator. Table 1 shows the results obtained from ten replicate runs of a genetic simulated annealing algorithm (GSA) each for 5,000 generations with a population size of 300. The GSA is based on the simple GA described earlier, but a single mutation follows each crossover operation, and the resulting offspring replaces a parent according the



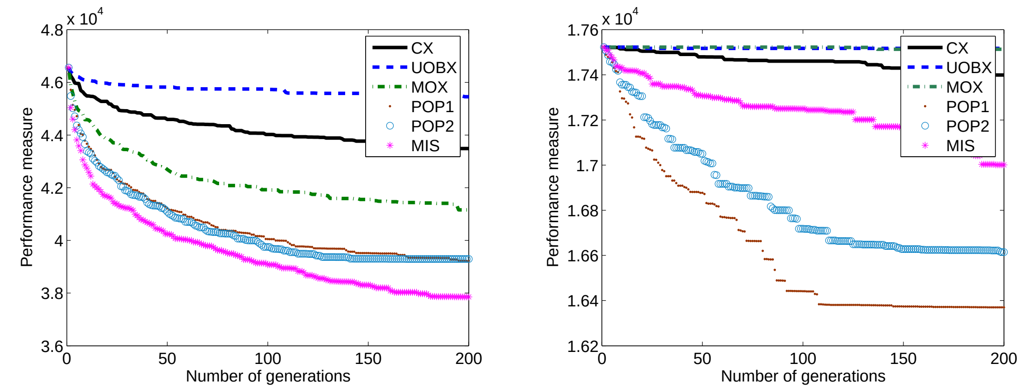
(a) DSJC500.5 (b) le450_15c

Fig. 4. Comparing order based crossovers with no sorting of independent sets and no local search



(a) DSJC500.5 (b) le450_15c

Fig. 5. Comparing order based crossovers with sorting of independent sets



(a) DSJC500.5 (b) le450_15c

Fig. 6. Comparing order based crossovers with sorting of independent sets and local search. Typical best colorings are 52 for DSJC500.5 and 26 for le450_15.c

standard simulated annealing formula. The chosen mutation consists of a simple inversion operation between two random cut points. MIS crossover is used for the DSJC instances and POP for the remainder. A GSA was chosen because it produced slightly more promising results than the other GA techniques tried so far by the author, although run times are long. However, this represents "work in progress" and there is much more to be done. In Table 1 results produced by the GSA are compared with those generated using a version of the algorithm with the crossover removed, but with mutation and local search intact. Results for iterated greedy and DSatur are also included for comparison. Runs of iterated greedy are replicated ten times, and the run times adjusted to match those of the GSA. Clearly, the GSA with crossover outperforms the mutation only version, reinforcing the valuable contribution made by the new crossover operators. The GSA also performs better than iterated greedy or DSatur on most of the instances. The final column in Table 1 gives the results reported by Galinier and Hao, and these are better than those produced by the GSA with the exception of le450_15c, where the results are matched.

Table 1. Comparison of the GSA with Other Approaches

Instance	Order Based GSA				Mutation GSA			Iterated Greedy			DSat	Best
	Time	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max		known
DSJC250.5	1618	**29**	29.5	30	31	31.2	32	**29**	29.5	30	37	*28*
DSJC500.5	6473	**50**	50.1	51	55	55.9	56	52	52.6	53	65	*48*
DSJC1000.5	25777	**86**	87.2	89	100	100.4	101	96	96.8	98	115	*83*
le450_15c	4007	*15*	15.1	16	25	25	25	23	23.5	24	23	*15*
le450_25c	4563	**29**	29.9	30	30	30	30	**29**	**29**	**29**	**29**	*26*
flat300_28	2081	**32**	32.2	33	35	35	35	**32**	33.1	34	42	*31*
flat1000_76	27204	**91**	92	93	99	99.6	100	95	95.5	96	114	*83*

5 Conclusions

The paper has presented two new order based crossover variations: *Merge Independent Sets (MIS)* and *Permutation One Point (POP)* for the GCP, and demonstrated their success in a simple genetic algorithm, comparing their performance with other crossovers on DIMACS benchmarks. The new crossovers appear to owe much of their success to an ability to respect color set boundaries, and this is made possible by utilizing some reordering heuristics taken from Culberson and Luo, [4]. In the experiments MIS seemed to work better than POP on problems where color classes vary in size, and POP proved more successful on the "flat" problems (le450_15c, le450_25c, flat300_28, and flat1000_76), which are specially formulated so that color class sizes are identical in the optimum solution. More extensive experiments showed that a genetic simulated annealing algorithm (GSA) worked much better with the new crossovers included than it did if they were excluded, and further, that the GSA is generally more effective than Culberson and Luo's iterated greedy algorithm, the source of the reordering heuristics used for MIS and POP. Thus, we have clear evidence that the crossovers provide much "added value" over and above mutation and local search. Future work will concentrate on improving the results further for the graph coloring problem and extending the approach to other set partitioning problems, initially concentrating on timetabling.

References

1. P. G. Anderson and D. Ashlock. Advances in ordered greed. 2004. Available from http://www.cs.rit.edu/˜pga/abstracts.php.
2. E. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2):266–280, 2002.
3. D. Costa, A. Hertz, and O. Dubuis. Embedding a sequential procedure within an evolutionary algorithm for coloring problems. *Journal of Heuristics*, 1:105–128, 1995.
4. J. Culberson and F. Luo. Exploring the k-colorable landscape with iterated greedy. In Johnson and Trick [11], pages 499–520.
5. D.Brélaz. New methods to color the vertices of graphs. *Communications of the ACM*, 24(4):251–256, 1979.
6. A. Eiben, J. V. der Hauw, and J. V. Hemert. Graph coloring with adaptive evolutionary algorithms. *Journal of Heuristics*, 4:25–46, 1998.
7. C. Fleurent and J. A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63:437–461, 1996.
8. P. Galinier and J. K. Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3(4):379–397, 1999.
9. C. A. Glass and A. Prügel-Bennett. A polynomially searchable exponential neighbourhood for graph colouring. *Journal of the Operational Research Society*, 56(3):324–330, 2005.
10. D. E. .Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
11. D. S. Johnson and M. A. Trick, editors. volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1996.
12. L.Davis. Order-based genetic algorithms and the graph coloring problem. In *Handbook of genetic algorithms*, chapter 6, pages 72–90. Van Nostrand Reinhold, New York, 1991.
13. D. Matula, G. Marble, and J. Isaacson. Graph coloring algorithms. In *Graph theory and computing*, pages 104–122. Academic Press, 1972.
14. C. Morgenstern. Distributed coloration neighborhood search. In Johnson and Trick [11], pages 335–358.
15. I. M. Oliver, D. J. Smith, and J. Holland. A study of permutation crossover operators on the traveling salesman problem. In *Genetic Algorithms and their Applications:Proceedings of the Second International Conference on Genetic Algorithms*, pages 224–230, 1987.
16. C. Valenzuela. A study of permutation operators for minimum span frequency assignment using an order based representation. *Journal of Heuristics*, 7(1):5–22, 2001. C.L. Valenzuela is now known as C. L. Mumford.
17. D. Welsh and M. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10:85–86, 1967.