Hill-Climbers, a Memetic Algorithm, and their Comparison on the Minimum Linear Arrangement Problem

Bryant A. Julstrom

Department of Computer Science
St. Cloud State University, St. Cloud, MN, 56301 USA
julstrom@eeyore.stcloudstate.edu

Abstract. Memetic algorithms combine evolutionary, population-based search and local, individual search, and the local search operator is sometimes stochastic hill-climbing. Stochastic hill-climbers alone can search effectively, so here we build a memetic algorithm by applying infrequent episodes of recombination to a population of them. Independent stochastic hill-climbers with and without these episodes are compared on eleven instances of the minimum linear arrangement problem, which seeks an ordering of a graph's vertices so as to minimize a sum over the graph's edges. Episodes of recombination consistently and significantly improve the performance of the stochastic hill-climbers, and the resulting algorithm can compete with recent, good heuristics for the problem, at least on the small instances addressed here.

1 Introduction

Among the heuristics applied to computationally difficult optimization problems are hill-climbers and memetic algorithms. A hill-climber subjects its one solution to a sequence of changes, incorporating those that improve the solution. When the changes are generated randomly, the hill-climber is said to be stochastic. Hill-climbers often perform well on optimization problems, and a population of n independent hill-climbers, as Section 2 describes, ascends up to n hills and will in general find better solutions than can one hill-climber alone.

Memetic algorithms [1] are hybrid search heuristics that combine evolutionary, population-based search and local, individual search. The most common form of memetic algorithm augments an evolutionary algorithm by applying local search to each offspring (e.g., [2], [3], [4], [5], and many more). A memetic algorithm's local search operator may be heuristic, as in the references just cited, but it may also be simple stochastic hill-climbing. Memetic algorithms of this design appear early (e.g., [6]) and continue to be both popular and effective (e.g., [7]).

Just as local search often improves the performance of an evolutionary algorithm, evolutionary operations such as recombination may improve the performance of a population of independent hill-climbers. Section 3 describes a memetic algorithm that, instead of applying local search to each offspring an EA generates, augments a population of stochastic hill-climbers with infrequent episodes of recombination.

Section 4 describes the minimum linear arrangement problem, in which, given an undirected graph $G = (V, E)$, we seek an arrangement $f : V \to \{1, 2, \ldots, |V|\}$ that minimizes the sum $\sum_{(u,v)\in E} |f(u) - f(v)|$. Implementations of independent stochastic hill-climbers and of hill-climbers augmented with episodes of recombination, as Section 5 describes, are compared on eleven small instances of the minimum linear arrangement problem, in tests that Section 6 presents. On the test instances and with the coding and operators the algorithms use, the inclusion of episodes of recombination consistently and significantly improves the performance of the hill-climbers.

2 A Population of Stochastic Hill-Climbers

Hill-climbing is a heuristic optimization strategy that begins with an initial solution to the target problem, then incrementally and monotonically improves it. One iteration of a hill-climber generates one or several neighbors of the current solution, then moves to the best of them if it improves on the current solution.

Hill-climbers are characterized by how they define neighboring solutions, how many neighbors they examine at each iteration, and how they choose those neighbors. An exhaustive hill-climber examines all the neighbors of the current solution at each iteration and halts when there is no better neighbor. Because the improvement at each step is always as large as possible, this technique is also called steepest ascent.

A stochastic hill-climber repeatedly generates one random neighbor of the current solution; whenever this neighbor is better, it replaces the current solution. The algorithm halts after it has examined a predetermined number of solutions or when the current solution has not improved for some number of iterations. This strategy conserves information by exploiting every improvement it finds, and it is in general far more efficient than exhaustive hill-climbing. On many problems, it competes effectively with more complicated heuristics like evolutionary algorithms (e.g., [8], [9], [10], [11]). Figure 1 summarizes stochastic hill-climbing.

```
s_o ← initial solution;
while ( not done )
    s_1 ← a random neighbor of s_o;
    if ( s_1 is better than s_o )
        s_o ← s_1;
report s_o;
```

Fig. 1. Stochastic hill-climbing. The algorithm repeatedly generates a random neighboring solution and moves to it if it is better

Regardless of the number of neighbors chosen in each iteration or how they are selected, a hill-climber can only climb the hill on which its initial solution falls, and this hill is in general not likely to contain a global optimum. This suggests maintaining a population of independent hill-climbers, each of which climbs its own hill. Among them, some will find themselves on higher hills, which they will climb to better solutions.

3 With Recombination

On a sufficiently complex problem, even a large collection of independent hill-climbers is likely to miss the hill (or hills) that contain globally optimal solution(s). Perhaps their performance can be improved by allowing them occasionally to exchange information.

In evolutionary algorithms, genotypes exchange genetic information via recombination operators, which generate offspring containing material from two parent genotypes. Here, we build a memetic algorithm by augmenting the process of independent hill-climbing with occasional episodes of recombination. In these episodes, each solution recombines with some or all of the other solutions, and the best of its offspring, if better, replaces the original solution. Then hill-climbing resumes. Figure 2 sketches the resulting algorithm. It is similar to Marchiori and Steenbeek's genetic local search algorithm [12] [13]; however, they apply genetic operators to a population of local optima generated by local search.

```
P ← initial solutions;
while ( not done )
   for every solution P[i] in P
      perform k iterations of hill-climbing;
   for every solution P[i] in P
      perform recombination with other solutions;
      P[i] ← the best solution among P[i] and its offspring;
report the best solution in P;
```

Fig. 2. Independent hill-climbing with episodes of recombination. P is the population of solutions and k is the number of hill-climbing iterations between episodes of recombination

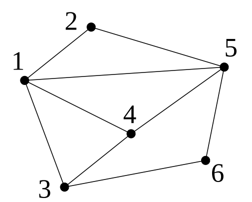
To maintain the spirit of independent hill-climbing, the episodes of recombination should occur only infrequently.

4 The Minimum Linear Arrangement Problem

Let $G = (V, E)$ be an undirected graph. A linear arrangement $f(\cdot)$ of G's vertices is a bijection $f : V \to \{1, 2, \ldots, |V|\}$, which imposes an ordering on the vertices, and a minimum linear arrangement (MinLA) is a linear arrangement that minimizes the following sum over G's edges:

$$\mathrm{LA}(G, f(\cdot)) = \sum_{(u,v)\in E} |f(u) - f(v)|.$$

Figure 3 shows a small graph, an arrangement of its vertices, and the computation of this sum for that arrangement.



v	$f(v)$
1	4
2	3
3	2
4	6
5	5
6	1

$\sum_{(u,v)\in E} |f(u) - f(v)| = |4 - 3|$
$+|4 - 2| + |4 - 6| + |4 - 5|$
$+|3 - 5| + |2 - 6| + |2 - 1|$
$+|6 - 5| + |5 - 1| = 18$

Fig. 3. A graph with $|V| = 6$ vertices and $|E| = 9$ edges, an arrangement $f(\cdot)$ of the graph's vertices, and the evaluation of the arrangement's sum

The MinLA problem is sometimes called the optimum linear ordering problem or the minimum linear ordering problem. It finds applications in a range of fields, including VLSI layout [14], computational biology [15], single-machine job scheduling [16] [17], graph drawing [18], and others. The problem is NP-hard for general graphs [19] and for bipartite graphs [20], though polynomial-time exact algorithms have been described for other special cases such as trees [21] [22], rooted trees [14], hypercubes [23], meshes [24], and outerplanar graphs [25]. Petit [26] [27] described an effective simulated annealing algorithm for the MinLA problem. Bar-Yehuda et al. [28] described a polynomial-time heuristic for it based on balanced decomposition trees.

5 Two Algorithms

A population of independent stochastic hill-climbers (HC) and hill-climbers with episodes of recombination (HC+X) pursue good solutions to instances of the minimum linear arrangement problem. They number the vertices of the target graph from 1 to $|V|$, and they represent candidate arrangements as permutations of the vertices' arrangement values: $s[i] = m$ represents $f(i) = m$. Both algorithms begin with a collection of random permutations.

The objective function, which the algorithms seek to minimize, is the sum $\mathrm{LA}(G, f(\cdot))$ of the absolute differences of arrangement values over the target graph's edges, as defined in Section 4. To find the sum associated with an arrangement, evaluation scans the edges (i, j) of the target graph and accumulates the values $|f(i) - f(j)| = |s[i] - s[j]|$. Note that evaluation's time is $O(|E|)$.

Let two permutations be neighbors if one can be turned into the other by exchanging two of its values. Neighbors are generated, then, by swapping two values in the current solution.

For HC+X, an appropriate recombination operator is Goldberg and Lingle's partially mapped crossover (PMX) [29]. This operation copies one of its two parent permutations into the offspring permutation, chooses a random segment within the permutations, and makes that segment in the offspring identical to those positions in the second parent by swapping values within the offspring. Note that this operator does not heuristically seek arrangements of lower evaluation. At long intervals, each hill-climbing solution $P[k]$ recombines with every other solution a small number m of times, and the best of these many offspring, if better that $P[k]$, replaces it when the recombination episode ends and independent hill-climbing resumes.

6 Tests

The independent hill-climbers (HC) and the hill-climbers with episodes of recombination (HC+X) were compared on eleven instances of the minimum linear arrangement problem. In the two smallest instances, the graphs consisted of 62 vertices with 144 edges and 65 vertices with 130 edges, respectively; the graph of the largest instance contained 250 vertices and 800 edges. Four of the instances are found in Petit's MinLA Repository[1]. They derive from problems in graph drawing, and their names begin with "gd". The remaining seven instances were generated for this study. Their edges were chosen randomly and they too are listed online[2]; their names begin with the letter 'j'. The first three columns of Table 1 below list all the instances and their numbers of vertices and edges. All of them are sparse, containing only a small proportion of all possible edges.

On an instance of the MinLA problem with n vertices, the number of independent hill-climbers (the population size) in both algorithms was $2n$. In HC+X, episodes of recombination occurred after every $20n$ iterations for each hill-climber, and there were $n/10$ such episodes, so the number of evaluations associated with hill-climbing in HC+X was $2n \times 20n \times n/10 = 4n^3$. In the recombination episodes, the number m of repeated recombinations was two, so each episode performed $2 \times 2n \times (2n-1) \approx 8n^2$ recombinations and evaluations and all the episodes together required $8n^2 \times (n/10)$ evaluations. Thus the total number of evaluations in a trial of HC+X was $4n^3 + 0.8n^3 = 4.8n^3$.

HC was allotted the same number of evaluations as HC+X, so each of its $2n$ simple hill-climbers performed $4.8n^3/(2n) = 2.4n^2$ iterations.

For example, on the MinLA instance `j100` with $|V| = 100$ vertices, the number of hill-climbers was 200. HC+X carried out 2000 iterations between recombination episodes, of which there were ten. In HC, each hill-climber climbed for 24000 iterations.

HC and HC+X were each run 50 independent times on each MinLA problem instance. Table 1 lists the results of these trials. In addition to the names and parameters of the instances, the table lists, for each algorithm applied to each instance, the value of the best arrangement found in the set of 50 trials and the

[1] `tracer.lsi.upc.es/minla/repository.php`

[2] `www.stcloudstate.edu/~julstrom/minla.html`

mean and standard deviation of the trials' results. We are interested in whether, on average, episodes of recombination improve the hill-climbers' performance, so the table also lists, for each instance, a t statistic derived from the two standard deviations and the difference of the two means as well as the number of degrees of freedom associated with that statistic.

Table 1. Results of trials of the independent stochastic hill-climbers, without and with episodes of recombination, on the eleven instances of the minimum linear arrangement problem.

Instance			Hill-Climbers			With Recombination			t-test	
Name	$\|V\|$	$\|E\|$	Best	$\overline{X}$	s	Best	$\overline{X}$	s	t	DF
gd95c	62	144	512	545.6	18.2	506	509.6	3.8	13.71	53
gd96c	65	130	525	534.8	6.2	521	530.9	4.7	3.58	91
j75	75	157	1520	1558.7	17.5	1500	1540.0	17.4	5.36	98
j100	100	250	3464	3526.3	29.8	3452	3502.2	25.6	4.37	96
gd96b	111	193	1422	1451.2	19.5	1416	1443.4	18.7	2.04	98
j125	125	230	2883	2959.9	33.4	2815	2890.4	36.2	9.98	97
j150	150	332	5820	5969.9	74.0	5701	5863.0	72.8	7.26	98
j175	175	363	6882	7074.4	84.5	6744	6899.6	75.4	10.91	97
gd96d	180	228	2639	2938.5	89.1	2477	2620.4	70.4	19.81	93
j200	200	558	15137	15473.0	107.7	15075	15294.0	107.8	8.31	98
j250	250	800	29510	29875.0	163.0	28969	29429.0	211.3	11.81	92

One observation stands out: The episodes of recombination do improve the performance of the independent hill-climbers. The best result returned by HC+X is always better than HC's best. More importantly, the mean of HC+X's results is always smaller than the mean of HC's results, and this difference, while not in general large, is always significant. Even the smallest t statistic—2.04 on gd96b—is significant at the 5% level in a one-sided t-test. We conclude that occasional episodes of recombination can improve the performance of a collection of otherwise independent hill-climbers.

Our goal was only to compare hill-climbers with and without episodes of recombination, but HC+X can compete with other heuristics for the minimum linear arrangement problem. Table 2 compares HC+X's results with those of Petit's simulated annealing [26] [27] and the algorithm of Bar-Yehuda et al. [28] on the four instances derived from graph drawing. The augmented hill-climbers' best results are as good as the other heuristics' on two of the instances and close on a third, though they do not compete effectively on the largest instance.

7 Conclusion

Memetic algorithms combine evolutionary and local search; the most common form of memetic algorithm supplements an evolutionary algorithm by applying

Table 2. Comparison of HC+X with the heuristics of Petit [26] [27] and Bar-Yehuda et al. [28] on the four graph-drawing-derived MinLA instances

Instance				Bar-Yehuda	HC+X	
Name	$\|V\|$	$\|E\|$	Petit	et al.	Best	$\overline{X}$
gd95c	62	144	509	506	506	509.6
gd96c	65	130	519	519	521	530.9
gd96b	111	193	1416	1422	1416	1443.4
gd96d	180	228	2393	2409	2477	2620.4

local search to each new solution, and the local search is sometimes carried out by stochastic hill-climbing. Here, conversely, a collection of independent stochastic hill-climbers has been augmented with occasional episodes of recombination to produce a different kind of memetic algorithm. Independent hill-climbers with and without episodes of recombination have been compared on the minimum linear arrangement problem.

A collection of stochastic hill-climbers represents candidate arrangements as permutations; neighbors differ by the exchange of two elements. In tests on eleven small instances of the MinLA problem, augmenting the hill-climbers with occasional episodes of partially mapped crossover yielded a consistently significant improvement in their performence. The augmented algorithm approaches the performance of other heuristics for the MinLA problem, at least on the small instances used here.

The independent hill-climbers might be improved in a variety of ways: having more of them, generating the initial solutions heuristically, using a heuristic neighbor operator. A new neighbor could be kept only when it is better than some number or all of the current solutions. Solutions could compete for survival, as in the "go with the winners" strategy of Aldous and Vazirani [30], where periodically the poorer solutions are moved to the positions of the better solutions.

The recombination-augmented hill-climbers can be similarly modified. Their performance might be improved by a different recombination operator, rearranging the application of recombination, or generating more solutions when recombining each pair. In a similar algorithm applied to graph coloring, Galinier and Hao [2] found that the number of hill-climbing steps between episodes of recombination strongly influenced the algorithm's performance; perhaps here those intervals should be shorter or longer.

Larger questions include how we might reduce the number of evaluations in HC+X so as to address larger problem instances, how effective recombination-augmented hill-climbers might be on other problems—that is, other triples of coding, objective function, and operators—and how we might identify problems on which such an algorithm will perform well.

References

1. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Caltech Concurrent Computation Program Report 826, California Institute of Technology (1989)
2. Galinier, P., Hao, J.K.: Hybrid evolutionary algorithms for graph coloring. Journal of Combinatorial Optimization **3** (1999) 379–397
3. Hodgson, R.J.W.: Memetic algorithms and the molecular geometry optimization problem. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC00, Piscataway, NJ, IEEE Press (2000) 625–632
4. Areibi, S.: Memetic algorithms for VLSI physical design: Implementation issues. In Wu, A.S., ed.: 2001 Genetic and Evolutionary Computation Conference Workshop Program, San Francisco, CA (2001) 140–145 July 7.
5. Kim, Y.H., Kwon, Y.K., Moon, B.R.: Problem-independent schema synthesis for genetic algorithms. In Cantú-Paz, E. et al., eds.: Genetic and Evolutionary Computation – GECCO 2003, Part I. Volume 2723 of LNCS., Berlin, Springer-Verlag (2003) 1112–1122
6. Suh, J.Y., van Gucht, D.: Incorporating heuristic information into genetic search. In Greffenstette, J.J., ed.: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ, Lawrence Erlbaum (1987) 100–107
7. Raidl, G.R., Ljubic, I.: Evolutionary local search for the edge-biconnectivity augmentation problem. Information Processing Letters **82** (2002) 39–45
8. Forrest, S., Mitchell, M.: Relative building-block fitness and the building-block hypothesis. In Whitley, L.D., ed.: Foundations of Genetic Algorithms 2, San Mateo, CA, Morgan Kaufmann (1993) 109–126
9. Lang, K.J.: Hill climbing beats genetic search on a boolean circuit synthesis of Koza's. In: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, Morgan Kaufmann (1995)
10. Moore, S.J., Sinclair, M.C.: Design of routing tables for a survivable military communications network using genetic algorithms. In: Proceedings of the 1999 Congress on Evolutionary Computation. Volume 2., Piscataway, NJ, IEEE Press (1999) 1788–1795
11. Julstrom, B.A.: A genetic algorithm and two hill-climbers for the minimum routing cost spanning tree problem. In Arabnia, H.R., Mun, Y., eds.: Proceedings of the International Conference on Artificial Intelligence. Volume III., CSREA Press (2002) 934–940
12. Marchiori, E., Steenbeek, A.: A genetic local search algorithm for random binary constraint satisfaction problems. In Carroll, J., Damiano, E., Haddad, H., Oppenheim, D., eds.: Applied Computing 2000: Proceedings of the 2000 ACM Symposium on Applied Computing, Volume 1. New York, ACM Press (2000) 458–462
13. Marchiori, E.: Genetic, iterated, and multistart local search for the maximum clique problem. In Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R., eds.: EvoWorkshops 2002. Number 2279 in LNCS, Berlin, Springer-Verlag (2002) 112–121
14. Adolphson, D., Hu, T.C.: Optimal linear ordering. SIAM Journal on Applied Mathematics **25** (1973) 403–423
15. Karp, R.M.: Mapping the genome: Some combinatorial problems arising in molecular biology. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing. (1993) 278–285

16. Adolphson, D.: Single machine job scheduling with precedence constraints. SIAM Journal on Computing **6** (1977) 40–54
17. Ravi, R., Agrawal, A., Klein, P.: Ordering problems approximated: Single-processor scheduling and interval graph completion. In Leach, M.R.J., Monien, B., eds.: 18th International Colloquium on Automata, Languages, and Programming. Volume 510 of LNCS., Berlin, Springer-Verlag (1991) 751–762
18. Pach, J., Shahrokhi, F., Szegedy, M.: Applications of the crossing number. Algorithmica **16** (1996) 111–117
19. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoretical Computer Science **1** (1976) 237–267
20. Even, S., Shiloach, Y.: NP-completeness of several arrangement problems. Technical Report 43, Department of Computer Science, Technion, Haifa, Israel (1975)
21. Shiloach, Y.: A minimum linear arrangement algorithm for undirected trees. SIAM Journal on Computing **8** (1970) 15–32
22. Chung, F.R.K.: On optimal linear arrangements of trees. Computers and Mathematics with Applications **10** (1984) 43–60
23. Harper, L.H.: Optimal assignments of numbers to vertices. Journal of the SIAM **12** (1964) 131–135
24. Mitchison, G., Durbin, R.: Optimal numberings on an $n \times n$ array. SIAM Journal on Algebraic and Discrete Methods **7** (1986) 571–582
25. Frederichson, G.N., Hambrusch, S.E.: Planar linear arrangements of outerplanar graphs. IEEE Transactions on Circuits and Systems **35** (1988) 323–332
26. Petit, J.: Approximation heuristics and benchmarking for the MinLA problem. Technical Report LSI-97-41-R, Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàticas (1997)
27. Petit, J.: Experiments on the minimum linear arrangement problem. Technical Report LSI-01-7-R, Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàticas (2001)
28. Bar-Yehuda, R., Even, G., Feldman, J., Naor, J.: Computing an optimal orientation of a balanced decomposition tree for linear arrangement problems. Journal of Graph Algorithms and Applications **5** (2001) 1–27
29. Goldberg, D.E., Lingle, Jr., R.: Alleles, loci, and the traveling salesman problem. In Greffenstette, J.J., ed.: Proceedings of the First International Conference on Genetic Algorithms, Hillsdale, NJ, Lawrence Erlbaum (1985) 154–159
30. Aldous, D., Vazirani, U.: "Go with the winners" algorithms. In: Proceedings of the 35th IEEE Symposium on the Foundations of Computer Science. (1994) 492–501