

An Evolutionary Algorithm for Some Cases of the Single-Source Constrained Plant Location Problem

Bryant A. Julstrom
Dept. of Computer Science, St. Cloud State University
St. Cloud, MN 56301 USA
julstrom@stcloudstate.edu

ABSTRACT

An evolutionary algorithm for some instances of the single-source capacitated plant location problem encodes candidate solutions in two permutations, one of plant locations and a second of customers, with an integer that indicates the number of open locations. A greedy decoder identifies the solution such a genotype represents, and the EA searches for good solutions using only selection and mutation. In tests on 36 problem instances, the EA is competitive with a recent algorithm, though two superficially promising heuristic extensions do not improve its performance. The results support the general effectiveness of permutation codings in EAs that search for optimum subsets.

Categories and Subject Descriptors: G.2.1 [Mathematics of Computing]: Discrete Mathematics—*Combinatorics*; I.2.8 [Problem Solving, Control Methods, and Search]: Heuristic Methods

General Terms: Algorithms

Keywords: Plant location, warehouse location, facility location, single-source, capacitated, permutation coding, evolutionary algorithm

1. INTRODUCTION

In the single-source capacitated plant location problem (SSCPLP), we are given n locations at which we may open plants, each with a fixed cost c_i and a capacity, and m customers, each with a service cost s_{ij} from each location and a demand. Each customer will be served from exactly one plant. We seek locations at which to open plants and an assignment of customers to them that minimizes the total cost without the demands of any plant's customers exceeding the plant's capacity. We consider instances of this problem in which no customer's demand is greater than the capacity of any plant. Note that the SSCPLP is NP-Hard.

A recent algorithm by Cortinhal and Captivo's [1], which they called the Fourth Approach (The first three were GAs.) maintains thirteen pairs of candidate solutions, organized into a trinary tree of three levels. It obtains initial solutions by Lagrangean relaxation and generates new solutions by local search and by applying crossover and mutation to current solutions.

An evolutionary algorithm for the problem encodes candidate solutions in two permutations, of plant locations and

of customers. A greedy decoder identifies the open locations and the assignment of customers to them that such a genotype represents. The EA applies mutation but not crossover. In trials on 36 instances of the problem, the EA performed about as well as the Fourth Approach of Cortinhal and Captivo. Two superficially promising heuristic extensions of the EA did not improve its performance.

2. THE EVOLUTIONARY ALGORITHM

In the EA, a genotype consists of a permutation of length n of the plant locations and a second permutation of length m of the customers. In addition, a genotype includes an integer ℓ ; the first ℓ locations in their permutation are open and thus may have customers assigned to them.

A greedy decoder identifies the solution that a genotype represents. It considers the customers in their permutation order, assigning each to the open location that can satisfy its demand at the lowest cost. If no location among the first ℓ can accommodate the customer, ℓ is incremented, indicating the opening of a plant, and the customer is assigned to that plant. Because every plant's capacity is at least as large as every customer's demand, this next plant will be able to handle the customer. Also, subsequent customers may now be assigned to the newly-opened plant. A genotype's fitness is the sum of the costs of serving the customers from their assigned plants and the fixed costs of those plants.

In early tests, a crossover operator did not advance the EA's search, so no crossover operator is used. Mutation swaps two randomly selected locations a small, random number of times and similarly swaps two randomly selected customers a small, random number of times. It also, with probability 0.2 in both cases, increments or decrements the number of included plant locations.

The coding and mutation were implemented in a generational evolutionary algorithm. The EA initializes its population with genotypes whose permutations are random and whose number ℓ of included plant locations is set to one; evaluation increases ℓ as it includes locations to satisfy customers' demands. The EA selects genotypes to be parents in 2-tournaments without replacement, and mutation generates each offspring. The EA is 1-elitist, and it runs through a fixed number of generations.

In the trials below, on SSCPLP instances with up to $n = 50$ plant locations and $m = 50$ customers, the EA's population contained $2(n+m)$ genotypes, mutation swapped one pair of plant locations and up to four customer pairs, and the EA ran through $5(n+m)$ generations. On instances with $n = 100$ locations and $m = 1000$ customers, the EA's

Table 1: Results of the Fourth Approach of Cortinhal and Captivo [1] and the EA on the 24 smaller instances of the SSCPLP. For each instance, the table lists its numbers n of plant locations and m of customers, percent error of the Fourth Approach (+%), and performance of the GA: the number of trials out of 40 that found the optimum (“Hits”), the best (Best+%) and mean (Mean+%) percent errors, and the standard deviation $s(\%)$ of the excess percentages.

Inst.	n	m	C&C	Evolutionary Algorithm			
			+%	Hits	Best+%	Mean+%	$s(\%)$
cap61	16	50	0.00	40	0.000	0.000	0.000
cap62	16	50	0.00	40	0.000	0.000	0.000
cap63	16	50	0.01	0	0.004	0.004	0.000
cap64	16	50	0.05	0	0.722	0.722	0.000
cap71	16	50	0.00	40	0.000	0.000	0.000
cap72	16	50	0.00	40	0.000	0.000	0.000
cap73	16	50	0.00	40	0.000	0.000	0.000
cap74	16	50	0.00	40	0.000	0.000	0.000
cap91	25	50	0.00	40	0.000	0.000	0.000
cap92	25	50	0.00	0	0.278	0.286	0.023
cap93	25	50	0.22	0	0.462	0.462	0.000
cap94	25	50	0.21	0	0.482	0.485	0.013
cap101	25	50	0.00	40	0.000	0.000	0.000
cap102	25	50	0.00	40	0.000	0.000	0.000
cap103	25	50	0.00	40	0.000	0.000	0.000
cap104	25	50	0.00	40	0.000	0.000	0.000
cap121	50	50	0.00	29	0.000	0.108	0.093
cap122	50	50	0.00	0	0.279	0.286	0.021
cap123	50	50	0.02	0	0.331	0.331	0.000
cap124	50	50	0.19	0	0.482	0.485	0.001
cap131	50	50	0.00	24	0.000	0.043	0.054
cap132	50	50	0.00	40	0.000	0.000	0.000
cap133	50	50	0.00	28	0.000	0.033	0.053
cap134	50	50	0.00	40	0.000	0.000	0.000

population contained 200 genotypes, mutation swapped one or two pairs of locations and up to ten pairs of customers, and the EA ran through 500 generations. The EA was implemented in C++ and executed on a Pentium 4 processor running at 2.53GHz under Red Hat Linux 9.0.

3. PERFORMANCE

The EA was run 40 independent times on each of 36 instances of the restricted SSCPLP. These instances are found in Beasley’s OR-Library¹ [2]. The smaller of them have $n = 16, 25,$ or 50 locations and $m = 50$ customers, and the larger have $n = 100$ locations and $m = 1000$ customers. Among the latter, there are only three sets of costs; we obtain twelve problem instances by assigning four different location capacities, uniform across the locations, to each set of costs. The values of optimum solutions of these instances are known, and the EA’s results were compared with those of the Fourth Approach, which was also tested on these instances [1].

On the smaller instances, the performance of the EA did not, with one exception, surpass that of the Fourth Ap-

¹<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>

Table 2: Results of the Fourth Approach of Cortinhal and Captivo [1] and the EA on the twelve larger ($n = 100$ and $m = 1000$) instances of the SSCPLP, as in Table 1. The table also lists the common plant capacity for each instance.

Inst.	Capacity	C&C	Evolutionary Algorithm			
		+%	Hits	Best+%	Mean+%	$s(\%)$
capa	8000	0.28	0	0.063	0.521	0.514
	10000	0.12	0	0.064	0.549	0.246
	12000	0.29	21	0.000	1.104	0.933
	14000	0.00	0	0.002	0.423	1.147
capb	5000	0.01	0	0.193	1.231	0.957
	6000	2.87	0	0.268	1.268	0.728
	7000	2.04	0	0.366	1.203	0.499
	8000	0.47	0	0.565	1.059	0.410
capc	5000	0.70	0	0.063	0.832	0.607
	5750	0.65	0	0.001	0.205	0.243
	6500	1.15	0	0.042	0.392	0.230
	7250	0.09	0	0.001	0.196	0.203

proach. Throughout, both algorithms performed well, and the differences between them were small. The worst performance was that of the EA on instance **cap64**, but its solution costs were, on average, always within 0.75% of optimum. The appearance of non-zero mean errors with zero standard deviations (**cap64**, **cap93**, and **cap123**) suggests that the EA is sometimes drawn into local optima.

On the larger instances, the EA fared better: on nine instances of the twelve, the EA’s best result was better than the Fourth Approach’s. On one instance, the EA found an optimum solution on 21 trials out of 40, and on five instances, the EA’s mean results were better than the Fourth Approach’s, though more often, the Fourth Approach’s results were better than the EA’s mean results. Overall, both again did well, with the Fourth Approach’s error never exceeding 2.9% and the EA’s mean error always less than 1.3%.

4. TWO FAILED HEURISTICS

Seeding the population with one or several genotypes in which the customers were arranged in decreasing order of demand did not improve the EA’s performance and on some instances made it worse. The interactions between the selection of locations and the assignments of customers to them are apparently too complex to be exploited by a simple ordering mechanism.

Similarly, reexamining customer assignments when new locations were opened did not change the EA’s performance. Once new locations have been opened, later evaluations of descendants of that genotype, will consider those locations; re-scanning is redundant.

5. REFERENCES

- [1] M. J. Cortinhal and M. E. Captivo. Genetic algorithms for the single source capacitated location problem. In M. G. C. Resende and J. P. de Sousa, editors, *Metaheuristics: Computer Decision-Making*, pages 187–216. Kluwer, Boston, 2004.
- [2] J. E. Beasley. Obtaining test problems via Internet. *Journal of Global Optimization*, 8:429–433, 1996.