Genetic algorithm with iterated local search for solving a location-routing problem

Joey Eremondi
Bas Geerts
Mathijs Stertefeld

About Paper

- Published in Elsevier 2011
- Cited by 18 papers
- Written at
    - FSEGS, Tunisia
    - LAMIH, France

Authors

Houda Derbel

- 5 papers
- 17 citations

Bassem Jarboui

- 59 papers
- 713 citations

Authors

Saïd Hanafi

- 133 papers
- 993 citations

Habib Chabchoub

- 83 papers
- 153 citations

Outline

- **Problem Definition**
- Approach Taken
- Experimental Results
- Conclusion and Discussion

The Location Routing Problem

- Undirected graph G = (V,E), costs on edges
- Nodes are either Depots or Customers
- Each customer has a demand
- Each depot has a cost and capacity
- Each depot has a vehicle of unlimited capacity, can take product to customers

The Location Routing Problem

- Want to find:
  - A subset S of all the depots
  - A route starting and ending at each depot in S
- such that
  - Every customer has their demand delivered to them
  - No depot gives out more than its supply
  - The combined cost of depots and routes is minimal

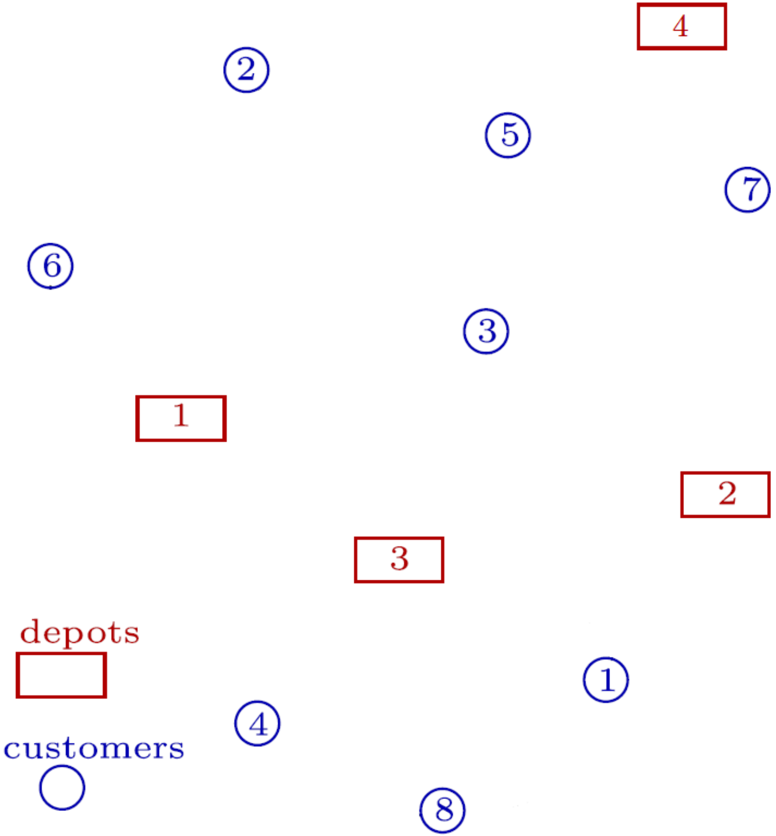
The Location Routing Problem

- Each potential solution has two vectors
  - *A*: the assignment vector
    - A[i] = k if customer i assigned to depot k
  - P: the permutation vector
    - Ordering of customers 1 to n
    - If customers i and j are assigned to depot k, visit i before j in the delivery route for k
- Some solutions might be equivalent

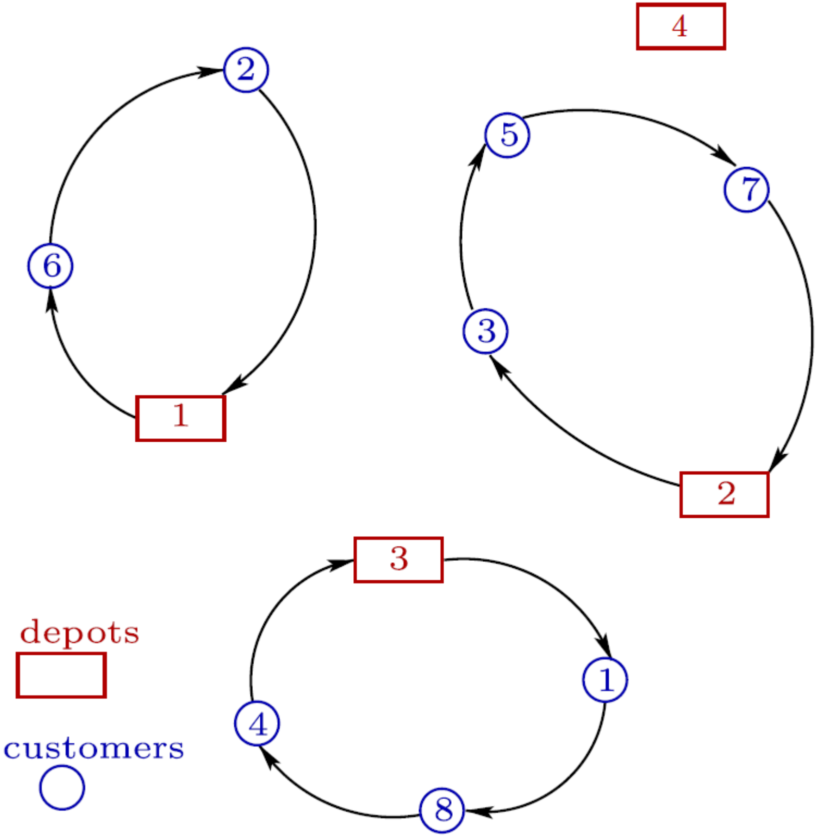
The Location Routing Problem

- Facility Location is NP-Hard
- Travelling-Salesman is NP-Hard
- Locating-routing requires solutions to both problems, so it is also NP-Hard

Example Problem



Example Solution



Outline

- Problem Definition
- **Approach Taken**
- Experimental Results
- Conclusion and Discussion

Hybrid Approach

- Use ILS to refine population of GA
- Given parents:
  - Generate a child using crossover and mutation
  - If fitness is within δ of the best so far, apply ILS on the child

Algorithm 1: **Hybrid GA&ILS high level overview**

```
s ← 40; ℙ_a ← 0.7; ℙ_p ← 0.9; δ ← 0.1; /* Global parameters */ ;
The GA algorithm high level code :
    FEVAL_best ← ∞;
    Initialize Pop /* First generation of individuals*/;
    Initialize the best fitness FEVAL_best ;
    while termination critrerion is not satisfied do
        SELECT individuals x_1 and x_2 from Pop following prob.
        Apply CROSSOVER to x_1 and x_2;
        Let x_new be the new obtained child;
        Apply MUTATION to x_new ;
        if FEVAL(x_new) < (1 + δ) · FEVAL_best then
            Apply ITERATIVE LOCAL SEARCH with x_new as an i
        Update the best fitness FEVAL_best;
        Apply REPLACEMENT to Pop;
```

Genetic Search: Selection

- According to probability distribution:
  - where *[k]* is the *k*th solution in descending order of its objective value
  - and *M* is the population size

$$\mathbb{P}([k]) = \frac{2k}{M(M+1)}$$

Genetic Search: Crossover

- Assignments *A*: simple one-point crossover
- *P* uses permutation-based crossover
- Point chosen from the first parent, permutation copied up until that point
- Elements of second parent inserted in order, skipping ones already added from first

Permutation-based crossover

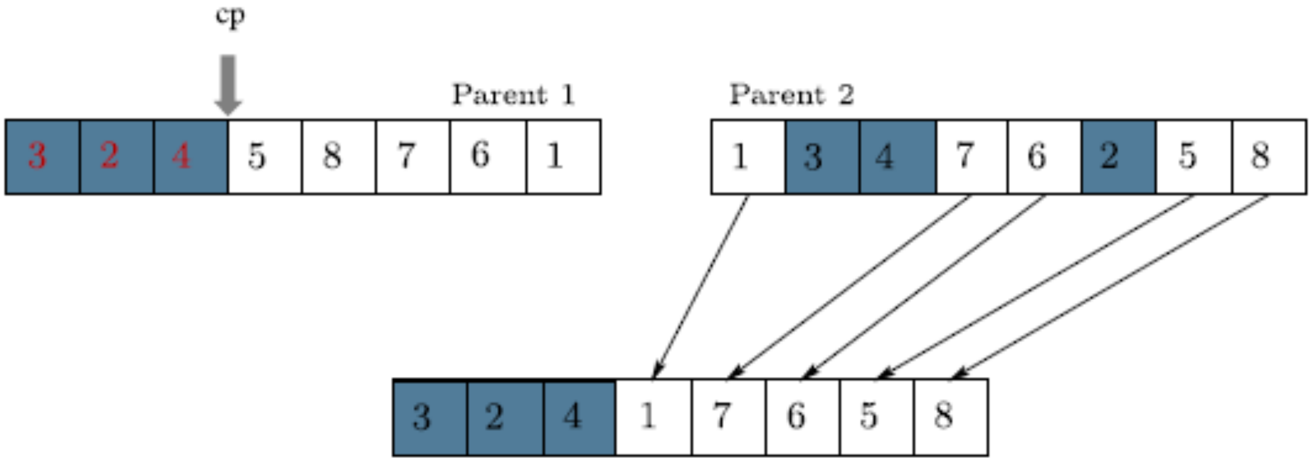


Fig. 2. Crossover operation for the permutation vector.

Genetic Search: Mutation

- *A* and *P* mutated separately
- Randomly move one customer to different depot
  - Allows potential depots to be added/removed from set of depots actually used
- Permutation: randomly select customer, re-insert into random position

Fitness function

- FEVAL(x) = COST(x) + PENALITY(x).
  - COST(x) = total cost of the LRP solution represented by individual x.
  - PENALTY(x) = a penalty on the violation of the capacity constraints

Fitness function

- More precisely:

$$\textsc{Penality}(x) = \sum_{j \in J} \alpha \cdot \max\{0, D_j(x) - b_j\}$$

  - where:
    - $D_j(x)$ is the total demand of customers assigned to depot *j* in solution *x*.
    - b_j is the maximal capacity of depot *j*.
    - α is a constant that reflects the degree of the penalty.

ILS: Neighbour Choice

- Use four separate neighbourhoods for each solution
  - Insertion move
  - Swap move

ILS: Neighbour Choice

- Sequentially improve an initial solution *x*
- Repeat until local optimum of the 4 structures of neighborhood is reached.

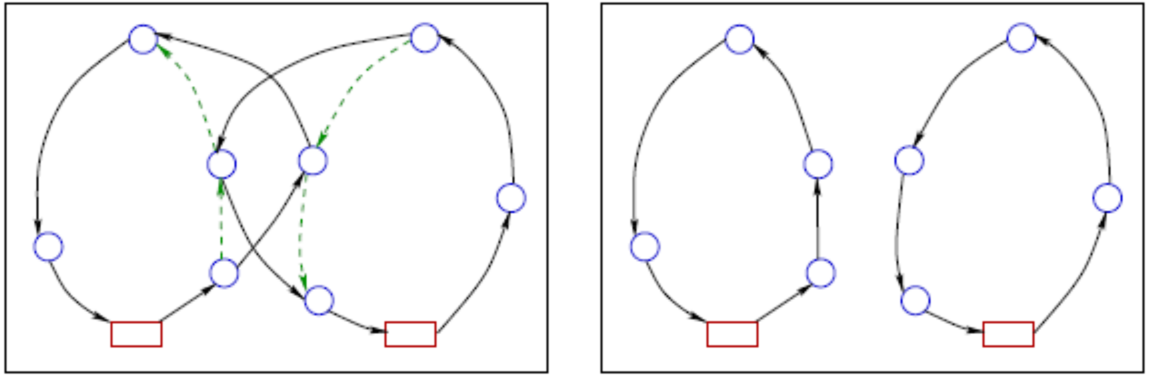
Algorithm 3: LOCAL SEARCH using neighborhoods $\mathcal{N}1$, $\mathcal{N}2$, $\mathcal{N}3$ and $\mathcal{N}4$

input : x : an initial solution

1 $x_1 \leftarrow$ FIRST IMPROVEMENT on x using neighborhood $\mathcal{N}1$;
2 $x_2 \leftarrow$ FIRST IMPROVEMENT on x_1 using neighborhood $\mathcal{N}2$;
3 $x_3 \leftarrow$ FIRST IMPROVEMENT on x_2 using neighborhood $\mathcal{N}3$;
4 $x_4 \leftarrow$ FIRST IMPROVEMENT on x_3 using neighborhood $\mathcal{N}4$;
5 **if** FEVAL(x_4) $<$ FEVAL(x_1) **then**
6     $x \leftarrow x_4$;
7   **Go to Line** 1;

Neighborhood N1

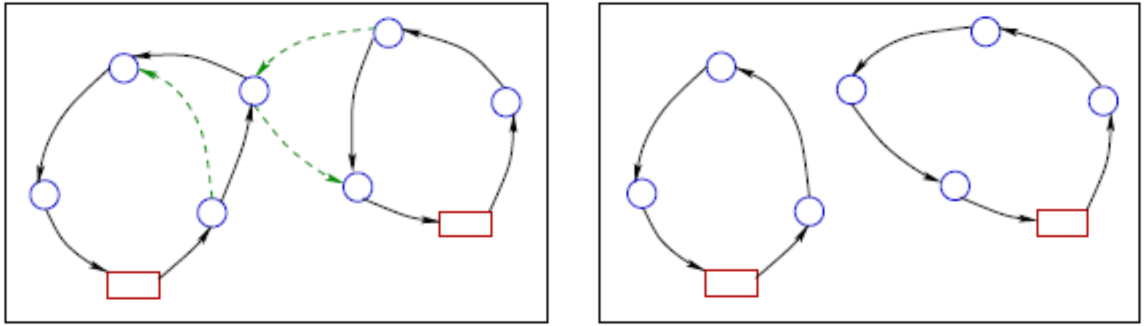
- Swap 2 random customers assigned to 2 different depots



(a) initial solution x (b) neighboring solution in $\mathcal{N}1(x)$

Neighborhood N2

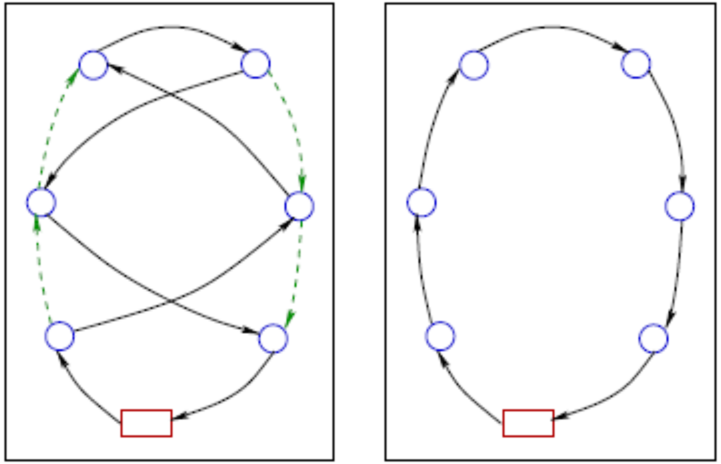
- Insert one customer from one route into another route



(a) initial solution x (b) neighboring solution in $\mathcal{N}2(x)$

Neighborhood N3

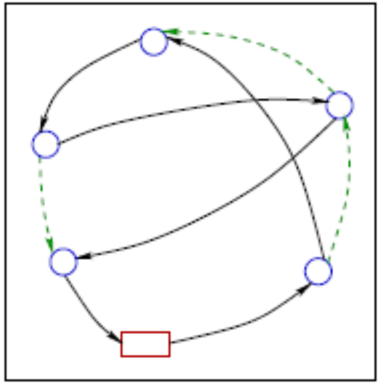
- Swap the position of 2 customers inside a route



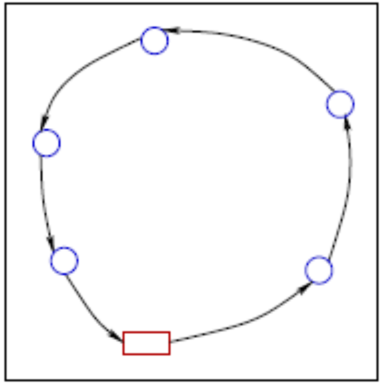
(a) initial solution x (b) neighboring solution in $\mathcal{N}3(x)$

Neighborhood N4

- Insert a customer between 2 other customers in the same route.



(a) initial solution x



(b) neighboring solution in $\mathcal{N}4(x)$

ILS: Perturbation Methods

- Opening closed depots gives us opportunities for different type of solutions
- Select an open depot at random
  - Remove the customers already assigned towards another depot (open or closed)
- This generates new kind of solutions by opening/closing some depots

Outline

- Problem Definition
- Approach Taken
- **Experimental Results**
- Conclusion and Discussion

The Experiment

- 5 data sets:
  - 5 facilities and {10, 20, 30} customers
  - 10 facilities and {20, 30} customers
- Vary ratio of total supply and total demand
- Vary average cost of opening a depot
- Compare with ILS and Tabu Search

Experiment Setup

- Coded in C
- Performed on a desktop computer
  - Windows XP
  - Intel Pentium IV - 3.2 GHz
  - 1 GB RAM

Experiment Results

Measured values:

- Average deviation of solution value relative to lower bound
- Running time of 10 instances

Experiment Results

Instances		p						m						g					
		ILS		GA&ILS		T.S		ILS		GA&ILS		T.S		ILS		GA&ILS		T.S	
		%gap	Time	%gap	Time	%gap	Time	%gap	Time	%gap	Time	%gap	Time	%gap	Time	%gap	Time	%gap	Time
*S*1	*a*	**0**	0.01	**0**	0.02	0.35	5.37	**0**	0.02	0	0.03	0.6	5.45	0.01	0.01	**0**	0.02	0.06	5.43
	b	**0**	0.00	**0**	0.02	0.04	4.89	**0**	0.01	**0**	0.02	0.35	5.10	**0**	0.01	**0**	0.02	0.04	5.04
	c	0.00	0.00	**0**	0.02	0.24	3.97	**0**	0.01	0	0.03	0.04	4.19	0	0.01	**0**	0.02	0.33	4.23
*S*2	*a*	**10.42**	0.41	**10.42**	0.40	10.79	22.7	7.66	0.83	**7.65**	0.92	7.97	24.94	**2.85**	2.28	**2.85**	0.64	2.94	25.91
	b	**11.58**	0.10	**11.58**	0.21	12.30	23.4	**8.88**	0.32	**8.88**	1.05	9.29	23.09	**4.02**	0.96	**4.02**	0.36	4.40	23.88
	c	**13.63**	0.12	**13.63**	0.22	13.69	17.47	**10.34**	0.90	**10.34**	1.00	10.44	17.42	**4.03**	0.2	**4.03**	0.55	4.08	17.37
*S*3	*a*	10.88	5.47	**10.28**	6.07	11.31	70.18	**6.28**	7.00	6.48	4.25	7.41	71.73	2.27	5.48	**2.26**	8.44	2.50	70.32
	b	11.35	4.95	**11.20**	4.44	12.33	62.78	**8.69**	3.66	8.78	7.62	10.30	64.88	3.21	2.36	**3.16**	2.55	3.79	65.53
	c	**13.26**	2.99	**13.26**	2.64	14.20	45.76	9.75	5.51	**9.55**	5.75	11.38	45.39	**3.75**	4.18	**3.75**	2.19	4.33	45.22
*M*2	*a*	21.75	1.90	**21.64**	4.71	22.01	90.55	14.15	2.40	**14.10**	5.28	14.90	91.14	6.08	3.18	**6.01**	7.30	6.13	90.72
	b	**27.37**	0.71	27.38	2.03	28.63	58.89	**17.17**	4.90	17.30	3.74	17.52	63.19	8.39	5.04	**8.30**	6.26	8.83	55.6
	c	29.60	2.85	**29.32**	2.43	30.71	40.30	**20.94**	1.67	21.02	3.67	22.98	38.13	**9.94**	4.54	**9.94**	4.87	11.54	35.60
*M*3	*a*	20.34	7.63	**19.43**	14.61	20.63	240.62	**12.79**	12.98	12.98	15.35	14.16	241.96	**6.63**	17.71	6.75	17.23	6.44	239.20
	b	**24.75**	4.82	24.82	18.07	26.65	139.08	16.97	6.58	**16.23**	9.33	17.45	163.97	**6.97**	11.42	8.18	15.23	8.77	133.81
	c	**26.98**	5.44	27.10	5.93	28.62	97.96	19.67	6.69	**19.65**	10.87	19.36	96.21	**9.11**	10.17	9.16	15.04	9.89	98.01

Solutions Found

- Found better solutions than Tabu in all tests
- Frequently found same or better solution than ILS
- Highest average deviation of 29.32%

Running Time

- Consistently faster than Tabu
- Ranged from slightly slower to much slower than ILS
- Longest running time is 18.07 seconds

Comparison

- Use of t-test
- Comparison between averages of two methods

$$\begin{cases} H_0 : & \alpha_1 = \alpha_2 \\ H_1 : & \alpha_1 < \alpha_2 \end{cases}$$

Comparison Results

H_0	H_1	t-value	p-value
ILS = T.S	ILS < T.S	−8.72	0.00
GA&ILS = T.S	GA&ILS < T.S	−9.83	0.00
GA&ILS = T.S	GA&ILS < T.S	−1.069	0.142

Outline

- Problem Definition
- Approach Taken
- Experimental Results
- **Conclusion and Discussion**

Conclusion

- Solution to two NP-Hard problems
- Combinations of GA and ILS
- Compared with best known methods
  - Higher accuracy
  - Better performance

Discussion

- Questions
- Comments