24th National Conference of HELORS

26-28 September 2013, NTUA, Athens, Greece

Generation of the exact Pareto set in multi-objective traveling salesman and set covering problems

Kostas FLORIOS

George MAVROTAS

Εργαστήριο Βιομηχανικής &
Ενεργειακής Οικονομίας
Σχολή Χημικών Μηχανικών
Εθνικό Μετσόβιο Πολυτεχνείο

Aim and scope

- Apply the Augmented ε-constraint method version 2
  - Mavrotas and Florios 2013
  - AUGMECON2
- Use two Multiobjective Integer Programming problems
  - Multiobjective Traveling Salesman Problem (2 obj.)
  - Multiobjective Set Covering Problem (2 obj.)
- Adapt the method for the Exact solution of MOTSP
  - Branch and Cut exact algorithm available in GAMS
  - AUGMECON2 in Fortran
- Compare to Multiobjective Metaheuristics for MOTSP
  - Lust and Teghem datasets and method
  - Paquete and Stüzle datasets and method
- Conclusions

LI
EE

Introduction

Multiple Objective Mathematical Programming (MOMP)

- More than one objective functions
- No unique optimal solution
- Most preferred among the Pareto optimal solutions (Steuer, 1986)

Classification of MOMP techniques in late '70s (Hwang and Masud, 1979)

- According to the phase in which the DM expresses own preferences
- A priori methods
- Interactive (or progressive) methods
- A posteriori (or generation) methods

Generation methods

- Less popular in the beginning of MOMP
- Due to increase of computer power they have become mainstream
- First, compute all (or a subset) POS
- Then, select the most preferred among the POS

Introduction (cont.)

Augmented ε-constraint method (AUGMECON) (Mavrotas, 2009)

- A Generation method
- Enhancement of conventional ε-constraint method (Miettinen, 1999)
- In general ε-constraint has advantages over weighted sum

The improved version of augmented ε-constraint method (AUGMECON2)

- Exploit slack/surplus variables
- Bypass coefficient
- Avoid many redundant iterations (especially with dense grids)

AUGMECON2 for MOIP problems (Mavrotas and Florios, 2013)

- Efficient Method in MOIP problems
- The method can be adjusted so that all POS can be generated
- MOCO problems treated as MOIP using this method
- Need for exact Pareto sets to benchmark MOMH in big datasets

LI EE

Model formulations

Multi-objective Integer Programming (MOIP)

(*Chinchuluun and Pardalos, 2007*)

$$\max \; Cx$$
$$s.t. \quad Ax \le b$$
$$x \ge 0 \qquad x \in \mathbb{Z}^n$$

Bi-objective Traveling Salesman Problem (BOTSP)

(*Lust and Teghem, 2010*)

(*Dantzig, Fulkerson, Johnson, 1954*)

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^{1} x_{ij}$$

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^{2} x_{ij} \qquad st$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,\dots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,\dots,n$$

$$\sum_{i\in Q}\sum_{j\in Q} x_{ij} \le |Q| - 1 \qquad for\ all\ Q \subseteq \{1,2,\dots,N\}\ and\ 2 \le |Q| \le N-1$$

$$x_{ij} \in \{0,1\} \quad i = 1,\dots,n, \; j = 1,\dots,n$$

$$vec\min \; z_k(\pi) = \sum_{i=1}^{N-1} c_{\pi(i),\pi(i+1)}^{k} + c_{\pi(N),\pi(1)}^{k}$$

L I
E E

Model formulations (cont.)

Multi-objective Integer Programming (MOIP)

(*Chinchuluun and Pardalos, 2007*)

$$\begin{aligned} \max \quad & Cx \\ s.t. \quad & Ax \leq b \\ & x \geq 0 \qquad x \in \mathbb{Z}^n \end{aligned}$$

Bi-objective Set Covering Problem (BOSCP)

(*Jaszkiewicz, 2004*)

$$\min \sum_{j=1}^{n} c_j^{(1)} x_j$$

$$\min \sum_{j=1}^{n} c_j^{(2)} x_j \qquad st$$

$$\sum_{j=1}^{n} t_{ij} x_j \geq 1 \qquad i = 1, \ldots, m$$

$$x_j \in \{0,1\} \qquad j = 1, \ldots, n$$

Methodology

The improved version of the augmented ε-constraint (AUGMECON2)

AUGMECON strengths related to ε-constraint

- Guarantee of Pareto optimality of the obtained solution in the payoff table as well as in the generation process
- Decreased solution time for problems with several (more than two) objective functions

AUGMECON2 strengths related to AUGMECON

- Introduction of bypass coefficient b
- Suitable for Multiobjective Integer Programming problems
- Remains an Exact method

Flowchart of AUGMECON2 for MOIP

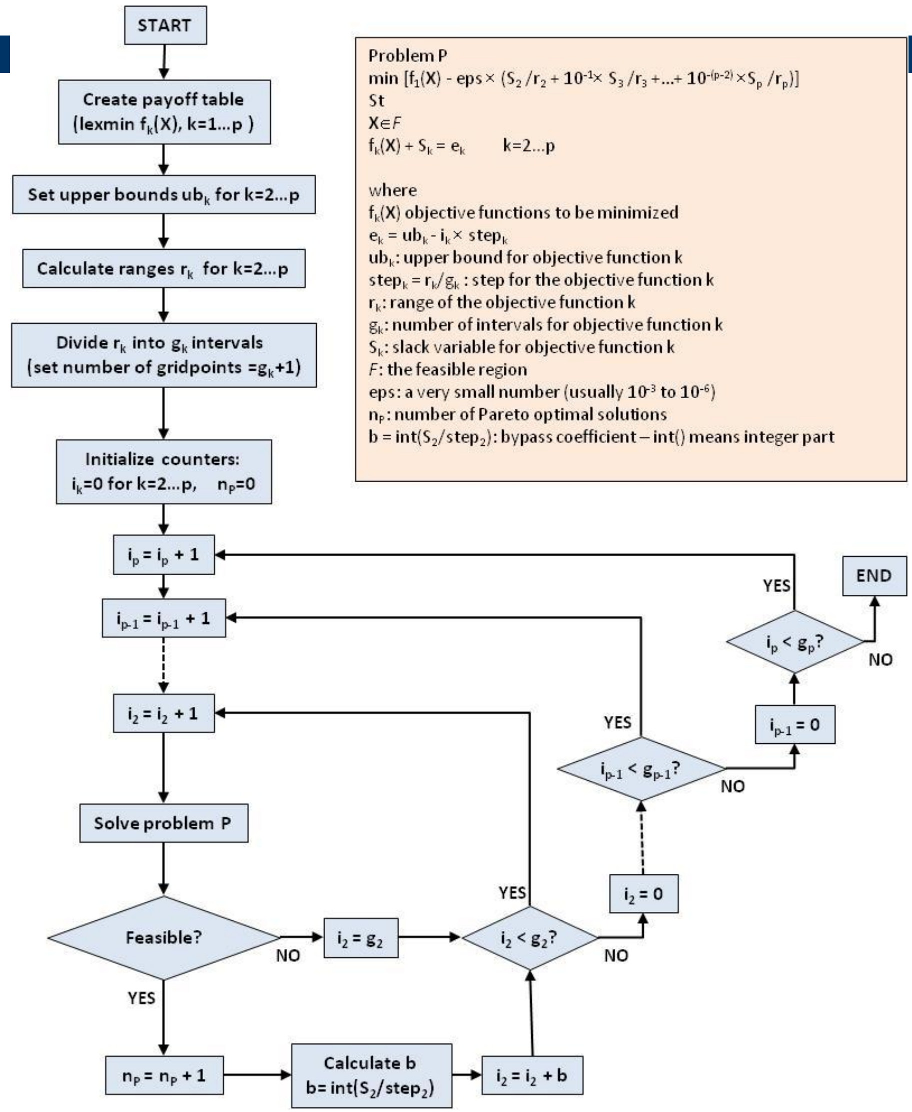


Figure 1 Flowchart of AUGMECON2 for MOIP

Adaptation of AUGMECON2 for MOIP

The size of the Pareto set is finite in MOIP problems

AUGMECON2 is able to generate the exact Pareto set for these problems

The only conditions are the following:

1. The objective function coefficients must be integer
2. The nadir points of the Pareto set must be known

The computational strategy for calculating the exact Pareto set in MOIP (AUGMECON2, coded in General Algebraic Modeling System, GAMS)

1. Calculate the objective function ranges r_k of the *p-1* objective functions, calculate or estimate upper bounds for the nadir points
2. Assume that r_k is integer. We select a unity step so that the number of grid points is exactly r_k+1 for every objective k=2,...,p
3. We apply AUGMECON2 and obtain the exact Pareto set. The unity step size and the calculation (or approximation with upper bounds) of the nadir point guarantee that no Pareto optimal solutions are left undiscovered

Adaptation of AUGMECON2 for MOTSP

An efficient formulation and solution method for single objective TSP: bchtsp.gms

Transform in order to handle 2 objectives (with ε-constraint subproblems)

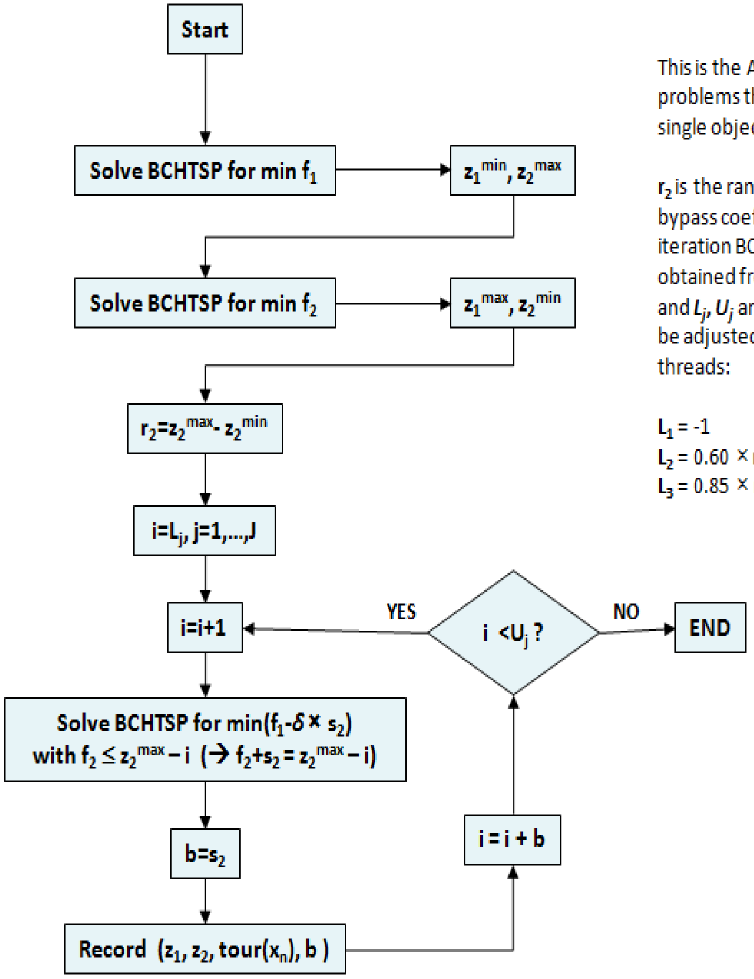
1. Input of the cost matrix c^2 for the second objective function
2. Input the nadir value z_2^{ub} and the ideal value z_2^{lb} of the second objective function
3. Solve bchtsp(η) for specific $\eta \in [0,1]$ externally defined by the AUGMECON2
4. Return solution information i.e. z_1, z_2, every time bchtsp(η) is called from AUGMECON2

The computational strategy for calculating the exact Pareto set in MOTSP

(AUGMECON2, coded in Intel Visual Fortran 11 and linked to GAMS using a system call)

```
r2= z2^ub - z2^lb , range of objective function 2
i=-1
stepsize=1
η_stepsize=stepsize/float(r2), a small number with 12 significant digits used to quantize η
c_POS=0
do while (i ≤ r2)
    i=i+1
    η=i·η_stepsize
    call bchtsp(η) implemented in GAMS returning, inter alia, the bypass coefficient b=s2
    i=i+b
    c_POS=c_POS+1
    write c_POS,z1,z2,i,b,ε2, CPU sec, tour in a diary file
Enddo
```

Flowchart of AUGMECON2 for MOTSP



This is the AUGMECON2 algorithm adapted for bi-objective TSP problems that use the BCHTSP code to solve the corresponding single objective problem.

r_2 is the range of the second objective function and **b** is the bypass coefficient as defined in Mavrotas and Florios [5]. In each iteration BCHTSP accepts as an argument the value of *b* that was obtained from the previous iteration. δ is a very small number and L_j, U_j are lower and upper bounds for the counter i that can be adjusted for parallelization. In the current case it holds for j=3 threads:

$L_1 = -1$ $U_1 = 0.60 \times r_2$
$L_2 = 0.60 \times r_2 - 1$ $U_2 = 0.85 \times r_2$
$L_3 = 0.85 \times r_2 - 1$ $U_3 = 1.00 \times r_2$

Figure 2 Flowchart of AUGMECON2 for MOTSP

L I
E E

Computational experiment

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

Table 1. The test bed of 16 datasets for two objective TSP

Lust's Instances	*Name*	*Paquete's Instances*	*Name*
L1	kroAB100	*P1*	euclAB100
L2	kroAC100	*P2*	euclCD100
L3	kroAD100	*P3*	euclEF100
L4	kroBC100	*P4*	randAB100
L5	kroBD100	*P5*	randCD100
L6	kroCD100	*P6*	randEF100
L7	euclAB100	*P7*	mixdGG100
L8	clusAB100	*P8*	mixdHH100
L9	randAB100	*P9*	mixdII100
L10	mixdGG100		

There are 3 parameters in the creation of Table 1 datasets:
a) Type of instances
b) Type of TSP and
c) number of cities, n

a) Euclidean type: L1-L6, L7, L8, P1-P3
Random matrix type: L9, P4-P6
Mixed type (of previous 2): L10, P7-P9
b) All studies instances = Symmetric TSP,
L1-L6 are Krolak/Felts/Nelson instances - with prefix kro in TSPLIB
L7-L10 and P1-P9 are DIMACS instances
c) n=100 for all 16 instances
(Note that the grey instances are common in the two testbeds)

web site https://sites.google.com/site/thibautlust/research/multiobjective-tsp Lust
web site http://eden.dei.uc.pt/~paquete/tsp Paquete
web site https://sites.google.com/site/kflorios/motsp Florios/Mavrotas

Computational experiment (cont.)

Biobjective Set Covering Problem (BOSCP), n=100-1000, m=10-200

Table 2. The benchmarks for the BOSCP

No	Model name	# Constraints	# Variables
1	*11*	10	100
2	*41*	40	200
3	*42*	40	400
4	*43*	40	200
5	*61*	60	600
6	*62*	60	600
7	*81*	80	800
8	*82*	80	800
9	*101*	100	1000
10	*102*	100	1000
11	*201*	200	1000

From Jaszkiewicz, 2004, Prins et al., 2006, Lust et al., 2011

The max number of 1s in a row i of matrix t is the parameter max-one

AUGMECON2 is compared to the adaptive ε-constraint procedure (Lust et al., 2011; Laumanns et al, 2006)

For every Number from 1 to 11, there exist 4 types of instances, namely A,B,C,D so there are totally 11×4=44 instances

web site http://xgandibleux.free.fr/MOCOlib/MOSCP.html
web site https://sites.google.com/site/kflorios/motsp

MOCOlib by X.Gandibleux
Florios/Mavrotas

L I
E E

Results and Discussion

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

Table 3. Augmecon2 statistics using the bchtsp model for 2-objective MOTSP (Lust, 10 datasets,)

Dataset	Pareto front size |PF*|	Models Solved (MS)	CPU time (h) in Parallel Processing Thread1 (h)	Thread2 (h)	Thread3 (h)
L1	3332	3372	47	45	71
L2	2458	2509	36	32	22
L3	2351	2370	15	20	26
L4	2752	2790	29	31	34
L5	2657	2705	25	28	27
L6	2044	2078	9	13	26
L7	1812	1839	19	11	19
L8	3036	3110	15	16	33
L9	1707	1718	7	9	25
L10	1848	1863	15	11	21

Table 4. Augmecon2 statistics using the bchtsp model for 2-objective MOTSP (Paquete, 9 datasets)

Dataset	|PF*|	MS	CPU time (h) in Parallel Processing Thread1 (h)	Thread2 (h)	Thread3 (h)
P1	1812	1839	19	11	19
P2	2268	2294	23	17	42
P3	2530	2559	13	22	28
P4	1707	1718	7	9	25
P5	1850	1868	13	15	19
P6	1882	1902	11	17	25
P7	1848	1863	15	11	21
P8	2108	2137	10	11	22
P9	1883	1906	13	16	20

(*) Hardware is a corei3 notebook capable of running 4 threads with Windows 7 32bit.

Results and Discussion (cont. 2/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

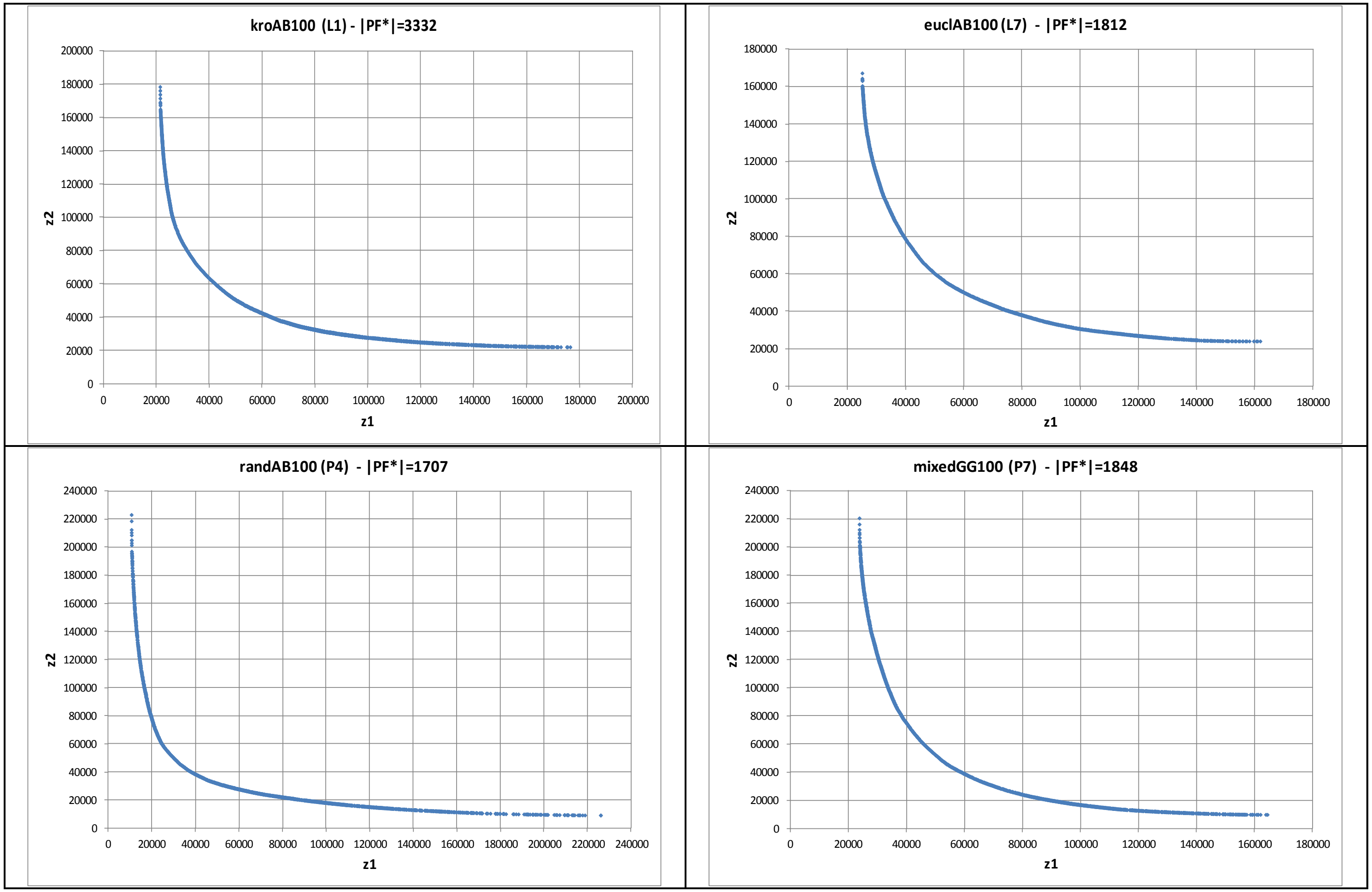


Figure 3. *Pareto fronts of kroAB100, euclAB100, randAB100, mixedGG100 from Lust & Paquete suites*

Results and Discussion (cont. 3/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

dataset	|PF*| exact	|PE| 2ppls	|D| 2ppls	|ND| 2ppls	C(2ppls,EPS) = |ND|/|PF*|
L1	3332	2640	988	1652	0.4958
L2	2458	2007	679	1328	0.5403
L3	2351	1885	730	1155	0.4913
L4	2752	2200	740	1460	0.5305
L5	2657	2058	579	1479	0.5566
L6	2044	1673	610	1063	0.5201
L7	1812	1397	502	895	0.4939
L8	3036	2557	878	1679	0.5530
L9	1707	663	266	397	0.2326
L10	1848	1011	376	635	0.3436

dataset	|PF*| exact	|PE| best known	|D| best known	|ND| best known	C(best,EPS) = |ND|/|PF*|
P1	1812	1719	76	1643	0.9067
P2	2268	2123	121	2002	0.8827
P3	2530	2387	68	2319	0.9166
P4	1707	1247	497	750	0.4394
P5	1850	1424	402	1022	0.5524
P6	1882	1287	611	676	0.3592
P7	1848	1644	145	1499	0.8111
P8	2108	1892	225	1667	0.7908
P9	1883	1724	132	1592	0.8455

$$C(A,B)=\frac{\left|\left\{b\in B \mid \exists a\in A : a\leq^{w} b\right\}\right|}{|B|}$$

• For L7=P1, Paquete finds 90.67% of the POS, and Lust's method only 49.39% of the POS.

• For L9=P4, Paquete finds 43.94% of the POS, and Lust's method only 23.26% of the POS.

• For L10=P7, Paquete finds 81.11% of the POS, and Lust's method only 34.36% of the POS

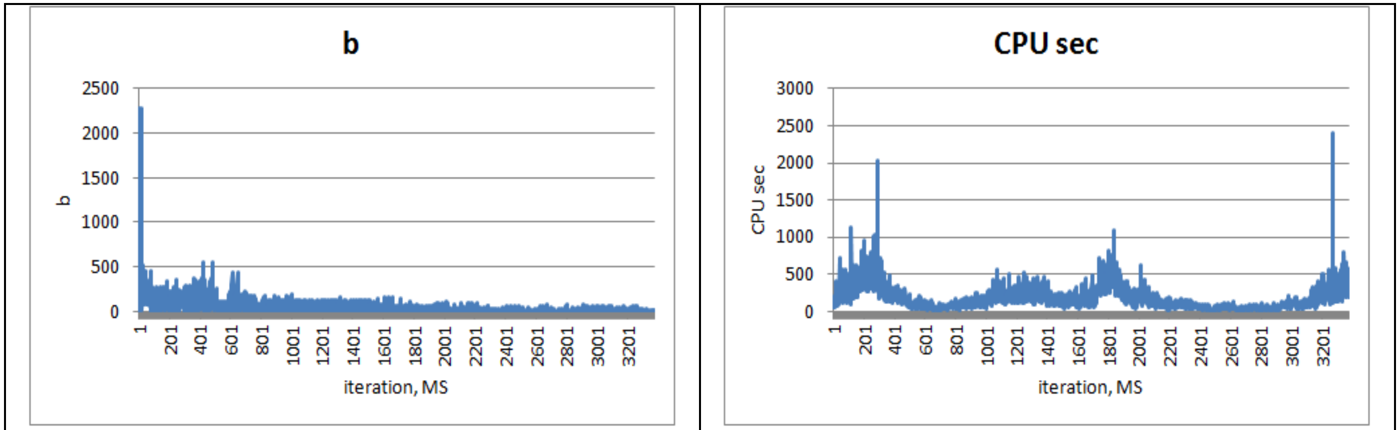
•LUST: Nevertheless, the coverage metric values are at most 50%-55%.
•PAQUETE: The coverage metric of 'best known' by Paquete for the Euclidean instances is very high, around 90%, which means a very good approximation of the EPS

Table 5. Coverage metric for two phase Pareto Local Search (2PPLS) of Lust and Teghem 2010 and Coverage metric for 'best non-dominated set found' of Paquete and Stützle 2009

LI EE

Results and Discussion (cont. 4/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities



(a) bypass coefficient, b (b) CPU sec

Figure 4. *Visualization of bypass coefficient, b, and CPU sec per iteration 'Models Soved' (MS) of our approach for the solution of kroAB100.*

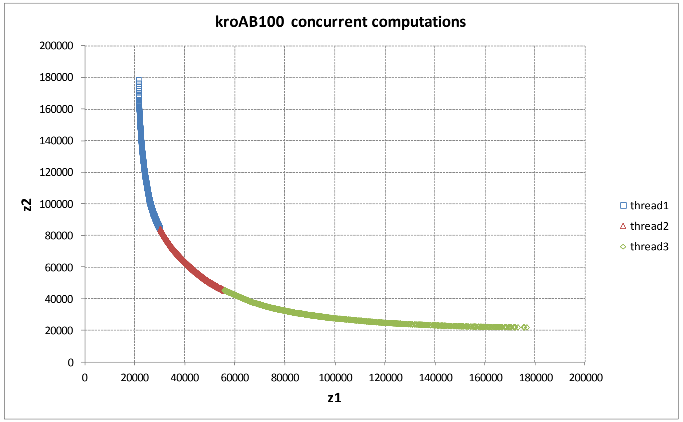


Figure 5. *AUGMECON2 concurrent computations in 3 threads for the solution of kroAB100*

Results and Discussion (cont. 5/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

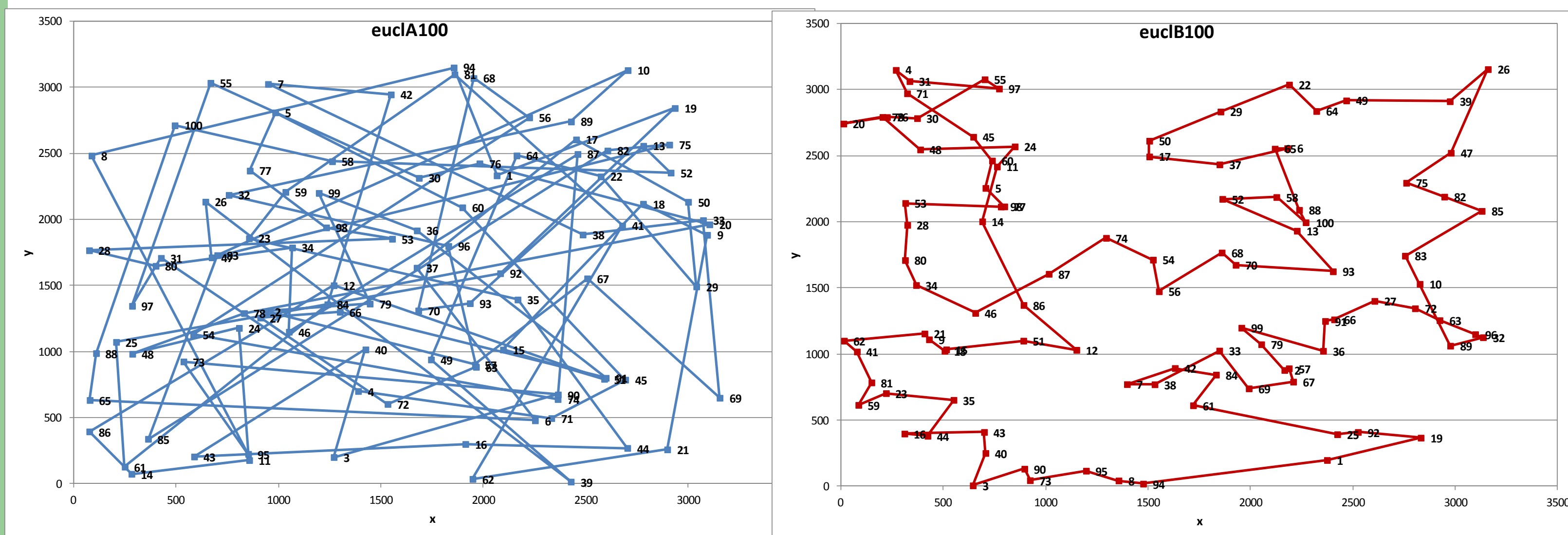
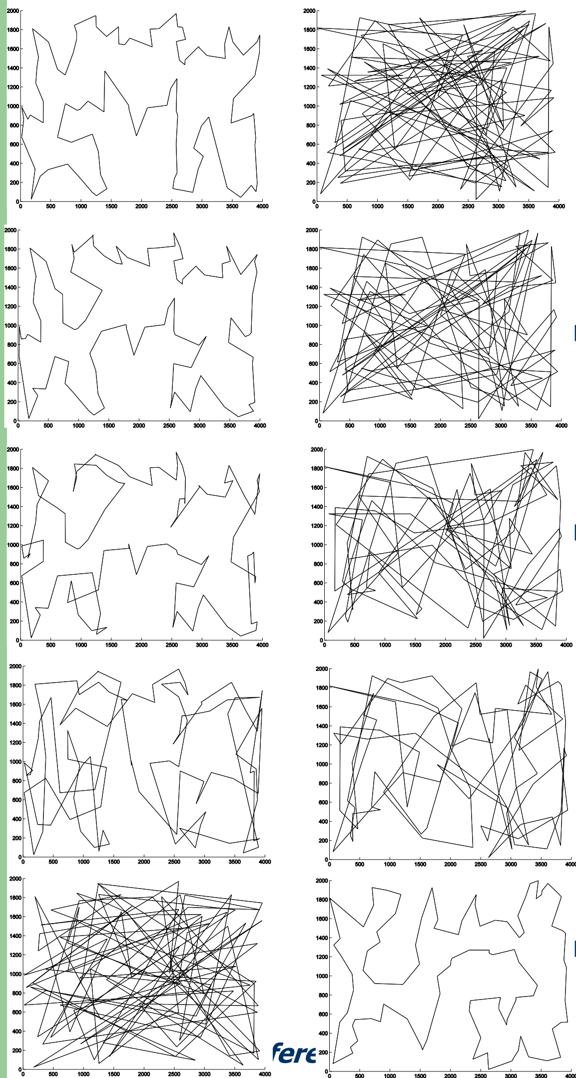


Figure 6. *Tours 1 and 2 of the Pareto Optimal Solution with η=0.975 for euclAB100*

Results and Discussion (cont. 6/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities



η = 0.00

η = 0.25

η = 0.50

η = 0.75

η = 1.00

Figure 7. Illustration of the computational trace for AUGMECON2 and kroAB100. On the left is kroA100 tour and on the right kroB100 tour. Observe the trade offs. At η=0.75 the tours are of similar quality

On the left: kroA100 tour
On the right: kroB100 tour

Play movie

$$\min \; z_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^1 x_{ij} \;\; \text{st}$$

$$z_2 = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^2 x_{ij}$$

$$z_2 + s_2 = \varepsilon_2$$

$$\varepsilon_2 = z_2^{ub} - \eta \cdot (z_2^{ub} - z_2^{lb})$$

$$x \in S$$

Results and Discussion (cont. 7/10)

Two objective Traveling Salesman Problem (MOTSP), 2 objectives, n=100 cities

Table 6 Percentage of Supported Efficient solutions over all POS in 16 datasets

	dataset	|PF*| exact	|SE| weighted sum	|SE|/|PF*| ratio
1	*L1*	3332	111	3.33%
2	*L2*	2458	106	4.31%
3	*L3*	2351	90	3.83%
4	*L4*	2752	114	4.14%
5	*L5*	2657	112	4.22%
6	*L6*	2044	98	4.79%
7	*L7*	1812	95	5.24%
8	*L8*	3036	109	3.59%
9	*L9*	1707	77	4.51%
10	*L10*	1848	98	5.30%
11	*P2*	2268	96	4.23%
12	*P3*	2530	100	3.95%
13	*P5*	1850	85	4.59%
14	*P6*	1882	89	4.73%
15	*P8*	2108	96	4.55%
16	*P9*	1883	92	4.89%
			Average	4.39%

It is impressive that, on average, only 4.39% are the POS that are supported. This actually means that when someone uses the weighting sum method for generating the POS in a MOTSP problem with N=100 cities, more than 95% of the true POS are left undiscovered

L I
E E

Results and Discussion (cont. 8/10)

Biobjective Set Covering Problem (BOSCP), n=100-1000, m=10-200

Table 7. Augmecon2 performance for BOSCP benchmarks No 1-11.

No	File	Type	# constraints	# variables	CPU sec	|PF*|	Models Solved
1	*11*	*A*	10	100	8.64	39	39
		B			6.26	43	44
		C			2.78	10	11
		D			1.45	5	7
2	*41*	*A*	40	200	18.01	107	108
		B			16.63	108	109
		C			7.96	24	25
		D			23.87	43	44
3	*42*	*A*	40	400	35.83	208	210
		B			52.04	276	280
		C			31.79	87	91
		D			7.14	15	16
4	*43*	*A*	40	200	12.80	46	47
		B			7.51	28	30
		C			3.78	13	14
		D			3.48	13	14

Results and Discussion (cont. 9/10)

Biobjective Set Covering Problem (BOSCP), n=100-1000, m=10-200

Table 7. Augmecon2 performance for BOSCP benchmarks No 1-11. (cont.)

No	File	Type	# constraints	# variables	CPU sec	|PF*|	Models Solved
5	*61*	*A*	60	600	83.66	257	261
		B			114.01	338	344
		C			18.49	28	31
		D			167.29	67	68
6	*62*	*A*	60	600	58.10	98	99
		B			60.20	99	100
		C			211.26	6	7
		D			134.17	38	45
7	*81*	*A*	80	800	148.66	424	430
		B			130.26	354	363
		C			7.76	14	17
		D			9.33	12	13
8	*82*	*A*	80	800	116.51	132	135
		B			38.16	88	94
		C			671.76	8	9
		D			1511.86	44	47

Results and Discussion (cont. 10/10)

Biobjective Set Covering Problem (BOSCP), n=100-1000, m=10-200

Table 7. Augmecon2 performance for BOSCP benchmarks No 1-11. (cont.)

No	File	Type	# constraints	# variables	CPU sec	|PF*|	Models Solved
9	*101*	*A*	100	1000	375.84	157	270
		B			225.50	141	142
		C			20933*	13	14
		D			3812	24	25
10	*102*	*A*	100	1000	104.76	83	87
		B			211.48	86	91
		C			2464*	14	15
		D			16724*	16	23
11	*201*	*A*	200	1000	6850*	274	282
		B			4278*	282	288
		C			dnt	dnt	dnt
		D			dnt	dnt	dnt

* 4 threads of CPLEX have been used

AUGMECON2	A	B	C	D	Average
100	8.64	6.26	2.78	1.45	4.78
200	15.41	12.07	5.87	13.68	11.76
400	35.83	52.04	31.79	7.14	31.70
600	70.88	87.11	114.88	150.73	105.90
800	132.59	84.21	339.76	760.60	329.29
1000	2443.5	1571.6	11698.0	10268.0	5597.86

Table 8 Performance of AUGMECON2 for BOSCP benchmarks No 1-11 averaged over types A-D.

LI
EE

Summary

- ☑ Apply the improved augmented epsilon constraint method
  - AUGMECON2
  - MOTSP – Multiobjective Traveling Salesman Problems
  - MOSCP – Multiobjective Set Covering Problems
  - Exact Pareto Set

- ☑ Adapt the AUGMECON2 method for the exact solution of MOTSP
  - An exact Branch and Cut for TSP – bchtsp in GAMS for TSP
  - Extend for the ε-constraint subproblem of MOTSP
  - Call iteratively from within AUGMECON2

- ☑ Use two MOTSP testbeds with 100 cities
  - Lust datasets - 9
  - Paquete datasets – 10
  - In total 16 datasets (3 are common)
  - Exact Pareto Set generated for first time with our approach

- ☑ MOTSP
  - 2 objectives: 100 cities solved, symmetric instances
  - euclidean, random matrix, mixed instances
  - AUGMECON2-BCHTSP calculates the Exact Pareto Set in 24 – 72 h.

Summary (cont.)

- ☑ MOTSP
  - Parallelization of the AUGMECON2-BCHTSP into 3 threads
  - $0\leq \eta \leq 0.60$ $0.60\leq \eta \leq 0.85$ $0.85\leq \eta \leq 1.00$
  - Split works fine, computational load of every thread balanced

- ☑ Evaluation of MOMHs
  - 2PPLS of Lust and Teghem
  - 'Best non dominated set found' by Paquete and Stüzle
  - Coverage(2PPLS,EPS) about 50-55% for Lust's method or lower
  - Coverage(best,EPS) about 90% for Paquete's ensemble results or lower
  - Best approximation achieved for euclidean datasets
  - Worst approximation achieved for random matrix datasets

- ☑ MOSCP
  - 42 out of 44 benchmarks of MOCOlib exactly solved in 24h limit
  - datasets 201a and 201b solved exactly for first time

- AUGMECON2 with BCHTSP solved 16 datasets of MOTSP with 2 objectives exactly – same datasets solved only approximately up to now

Future research considers parallelization of AUGMECON2 with Open-MP (in Fortran) and study of alternative bypass rules not only for the innermost loop (for 3 objective problems)

Selected references

- Hwang CL, Masud A. Multiple Objective Decision Making. Methods and Applications: A state of the art survey, Lecture Notes in Economics and Mathematical Systems Vol. 164. Berlin: Springer-Verlag; 1979.
- Mavrotas G, Florios K. An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. Applied Mathematics and Computation 2013;219:9652-69.
- Lust T, Teghem J. The multiobjective traveling salesman problem: A survey and a new approach. Studies in Computational Intelligence 2010;272:119-41.
- Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling-salesman problem. Journal of the Operations Research Society of America 1954;2:393-410.
- Mavrotas G., 2009. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems, *Applied Mathematics and Computation,* Vol. 213, No. 2, pp.455-465.
- Hansen MP. Use of substitute scalarizing functions to guide a local search based method: The case of moTSP. Journal of Heuristics 2000;6:419-31.
- Jaszkiewicz A. Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research 2002;137:50-71.
- Paquete L, Stützle T. Design and analysis of stochastic local search for the multiobjective traveling salesman problem. Computers & Operations Research 2009;36:2619-31.
- Lust T, Teghem J. Two-phase Pareto local search for the biobjective traveling salesman problem. Journal of Heuristics 2010;16:475-510.
- Jaszkiewicz A. A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the Pareto memetic algorithm. Annals of Operations Research 2004;131:135-58.
- Applegate D, Bixby R, Chvátal V, Cook W. On the solution of travelling salesman problems. Documenta Mathematica 1998;3:645-56.
- Bussieck MR. Introduction to GAMS Branch-and-Cut Facility. Technical report, GAMS Development Corp., 2003. http://www.gams.com/docs/bch.htm
- GAMS Corp., Traveling Salesman problem with BCH, http://www.gams.com/modlib/libhtml/bchtsp.htm.

Thank you

Hardware and Software

- This paper
  - AUGMECON2-BCHTSP: present paper
    - S/W= GAMS 23.5, CPLEX 12.2; Intel Visual Fortran 11; Windows 7 32bit
    - H/W= Core i3, 2.13GHz, 3GB RAM

LI
EE

Hardware LINPACK benchmarks this paper

	Computer	"LINPACK Benchmark" OS/Compiler	n=100 Mflop/s	"TPP" Best Effort n=1000 Mflop/s	"Theoritical Peak" Mflop
This paper					
4 threads benchmarks					
Section MOTSP, 2obj	Intel core i3 M330, 2.13GHz with 4GB RAM	Windows 7 32-bit	2406	7528	
Section BOSCP	Intel core i5 M520, 2.40GHz with 4GB RAM	Windows 7 32-bit	2943	7757	
1 thread benchmarks					
Section MOTSP, 2obj	Intel core i3 M330, 2.13GHz with 4GB RAM	Windows 7 32-bit	2993	5935	
Section BOSCP	Intel core i5 M520, 2.40GHz with 4GB RAM	Windows 7 32-bit	4016	7888	