A web-based solution platform for Multi-objective Integer Programs

Banu Lokman, Gökhan Ceyhan, Murat Köksalan

Middle East Technical University, Department of Industrial Engineering, Ankara

supported
by
The Scientific and Technological Research Council of Turkey
(Project No: 215M844)

July 2017

OUTLINE

- Introduction
- The Algorithms
- A web-based solution platform for Multi-objective Integer Programs – **under construction**
- Conclusions and Future Work

INTRODUCTION – Problem Definition

$$"Max" \left\{ z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_q(\mathbf{x}) \right\}$$

subject to

$$\mathbf{x} \in \mathbf{X}$$

where

z_i: the i^{th} objective function

$\mathbf{x} \in \mathbb{Z}^l$: decision vector

$\mathbf{X}$: solution (decision) space

q : the number of objectives

INTRODUCTION – Definition

$x_2 \in X$ is said to **dominate** $x_1 \in X$ and $z(x_2)$ is said to dominate $z(x_1)$ if

$$z_i(x_1) \leq z_i(x_2) \text{ for all } i = 1,\ldots, q$$

$$z_i(x_1) < z_i(x_2) \text{ for at least one } i.$$

A feasible solution $x \in X$ is called **efficient**, if there is **no other x'** $\in X$ such that

$$z_i(x) \leq z_i(x') \text{ for all } i = 1,\ldots, q$$

$$z_i(x) < z_i(x') \text{ for at least one } i.$$

If x is efficient, z(x) is called **nondominated** point.

INTRODUCTION – MOIP and MOCO

- Multi-objective Integer Programs (MOIPs) have many applications in real life:
  - ✓ Budgeting Problems
  - ✓ Network Design Problems
  - ✓ Routing Problems
  - ✓ Location and Hub-Location Problems
  - ✓ Scheduling Problems
- Finding nondominated points is typically hard.
- Ehrgott and Gandibleux (2000) review Multi-objective Combinatorial Optimization (MOCO) Problems, special MOIPs.
- X is discrete and "large".
- Grows fast with problem size

INTRODUCTION – Literature Review

Studies to generate all nondominated points

- Sylva and Crema (2004) solves sequence of progressively more constrained single-objective integer problems.
- Tenfelde-Podehl (2003), Dhaenens et al. (2010) and Przybylski et al. (2010), propose methods based on **two-phase method**.
- Laumanns et al. (2006) and Özlen and Azizoglu (2009) utilizes the **epsilon-constraint method**.
- Recently, more efficient algorithms developed by:

  Lokman and Koksalan (2013), Kırlık and Sayın (2014), Özlen et al. (2014), Dächert and Klamroth (2015), Boland et al. (2015, 2016).

where the properties of the epsilon-constraint method are used, and **the search region is decomposed**.

INTRODUCTION – Literature Review

Sylva and Crema (2004)

$(P_{\lambda(n)})$

$$Max \sum_{i=1}^{q} \lambda_i z_i(x)$$

subject to

$$z_i(x) \geq \left(z_i^k(x)+1\right)t_{ik} - M\left(1-t_{ik}\right) \ \forall i \ \ \forall k$$

$$\sum_{i=1}^{q} t_{ik} \geq 1 \ \ \forall k$$

$$t_{ik} \in \{0,1\} \quad k=1,\ldots,n \quad i=1,2,\ldots,q$$

$$x \in X$$

n(q+1) additional constraints and nq binary variables to guarantee a new nondominated point different from any of the existing ones.

INTRODUCTION – Literature Review

Lokman and Köksalan (2013)

$(P_{\lambda(n)})$

$$Max\ z_i(x) + \sum_{\substack{i=1 \\ i \neq m}}^{q} \rho_i z_i(x)$$

subject to

$$z_i(x) \geq \left(z_i^k(x) + 1\right) t_{ik} - M\left(1 - t_{ik}\right) \quad \forall i \neq m \quad \forall k$$

$$\sum_{\substack{i=1 \\ i \neq m}}^{q} t_{ik} \geq 1 \quad \forall k$$

$$t_{ik} \in \{0,1\} \quad k = 1,\ldots,n \quad i = 1,2,\ldots,q \quad i \neq m$$

$$x \in X$$

n(q) additional constraints
and
n(q-1) binary variables
to guarantee a new nondominated point different from any of the existing ones.

INTRODUCTION – Literature Review

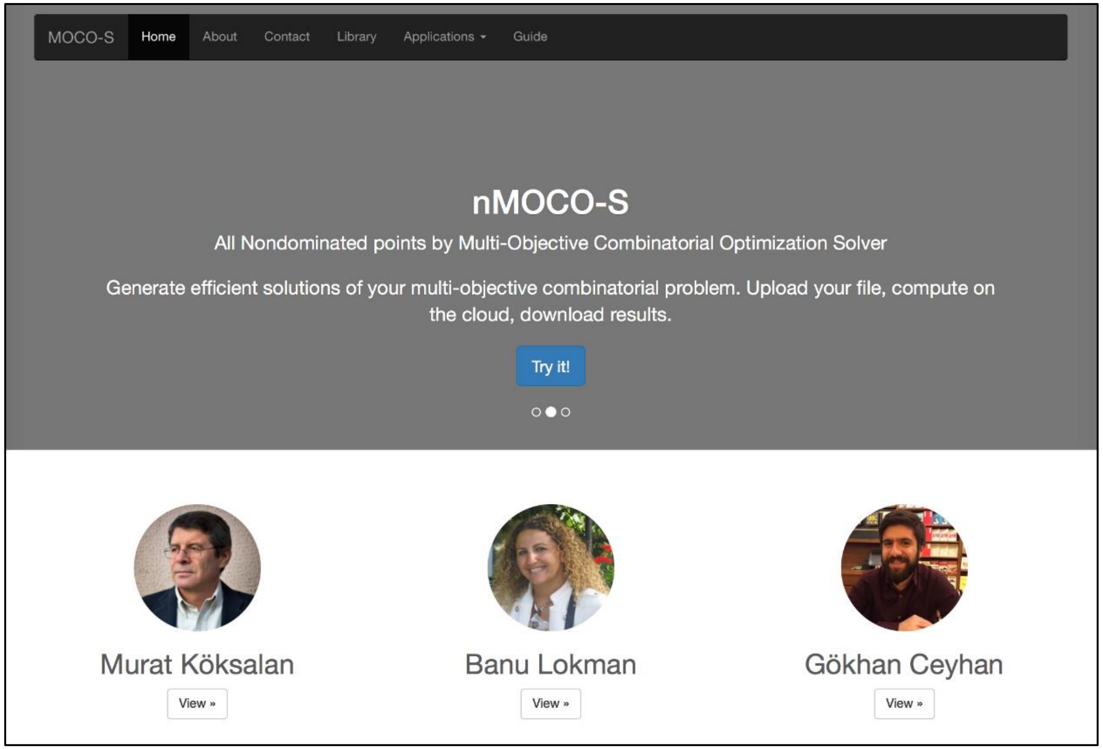
Lokman and Köksalan (2013)

- Partitions the feasible space into subspaces.
- Employs a search procedure.
- Solves a number of models by imposing bounds on the objectives rather than adding additional constraints or binary variables.

A web-based solution platform

A web-based solution platform to generate

- Nondominated points (all or regional)
- Representative nondominated set with desired quality level

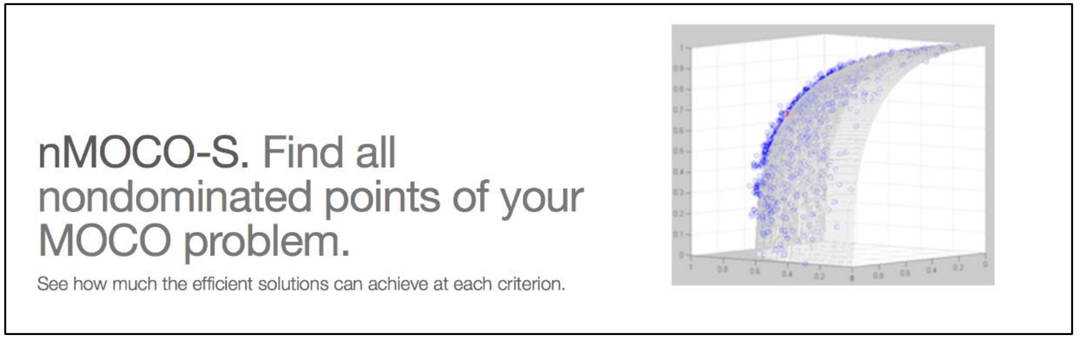


A web-based solution platform

www.onlinemoco.com/MOIP/

- In development phase.
- Applications (under construction):
  - **nMOCO-S:** Finds all nondominated points.
  - **rMOCO-S:** Finds a representative set of nondominated points
  - **iMOCO-S:** Integrates user preferences
  - **libMOCO-S:** Collects a variety of MOCO instances.

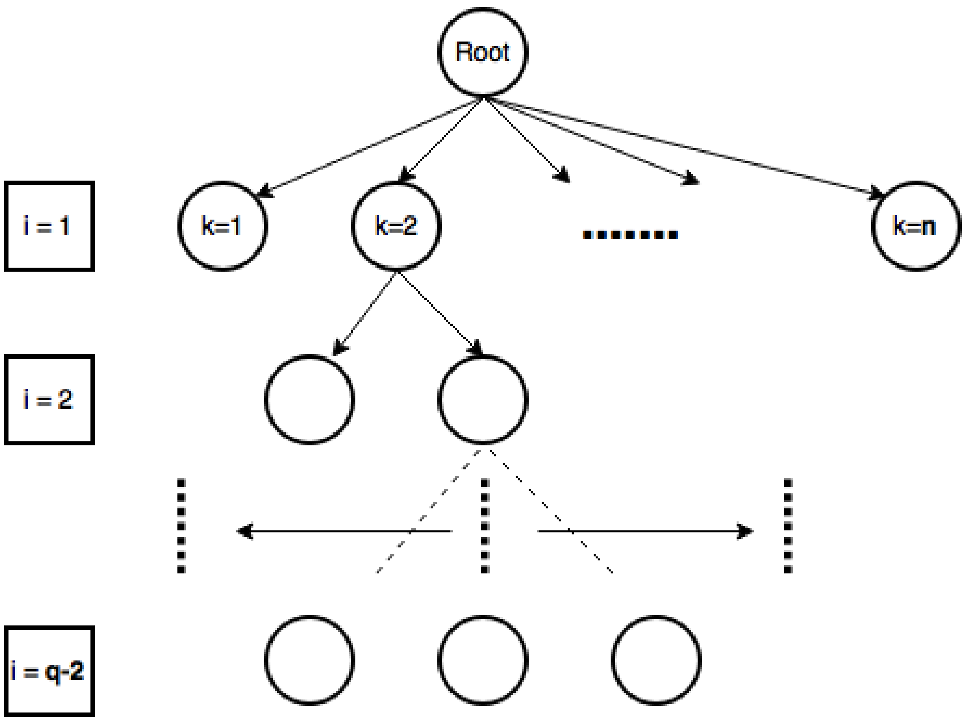
A web-based solution platform - nMOCO-S



- Generates all nondominated points for MOIPs.
- Based on Lokman and Köksalan (2013).
- **The idea:** The search region is partitioned and reduced progressively by removing the regions that are dominated by previously found nondominated points.

A web-based solution platform - nMOCO-S

The Approach - MOIP TREE



- **N-ary tree** structure
- **Tree height** = number of objectives - 2 = **q-2**
- **Number of child nodes of a node** = as many as the number of already generated nondominated points = **n**

A web-based solution platform - nMOCO-S

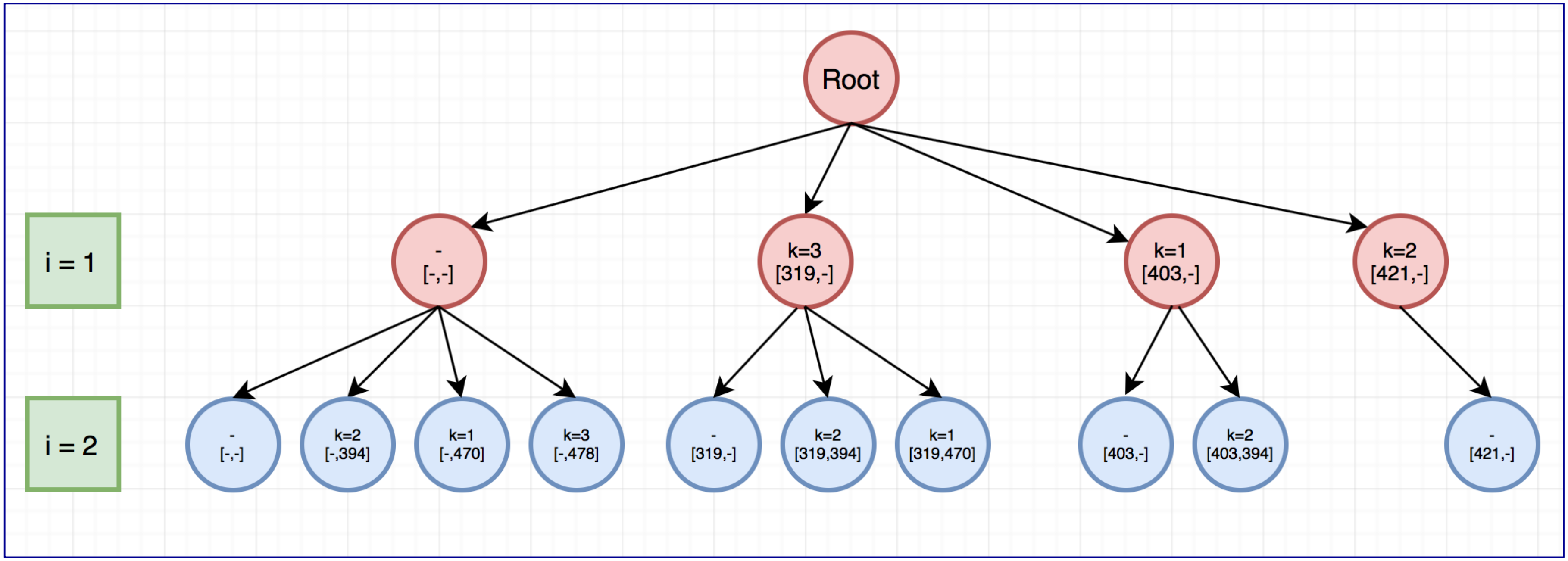
The Approach - Demo: Tree creation and update

Let's create the initial tree with the following three nondominated points in the four dimensional criterion space.

Points	z_1	z_2	z_3	z_4
1	402	469	521	324
2	420	393	508	452
3	318	477	487	521

A web-based solution platform - nMOCO-S

The Approach - Demo: Initial tree

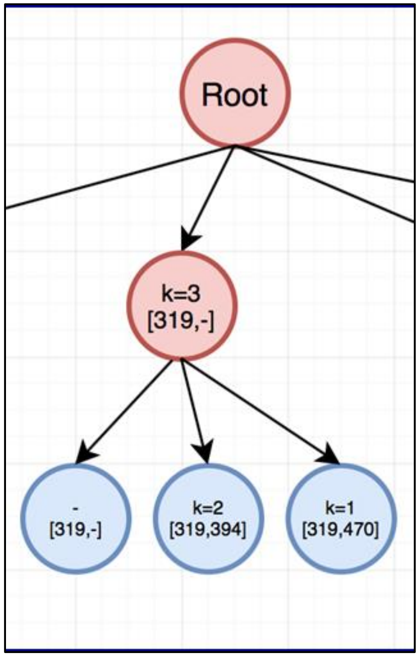


At leaf nodes: compute the bounds at criterion (q-1) and optimize **q**th obj. function.

A web-based solution platform - nMOCO-S

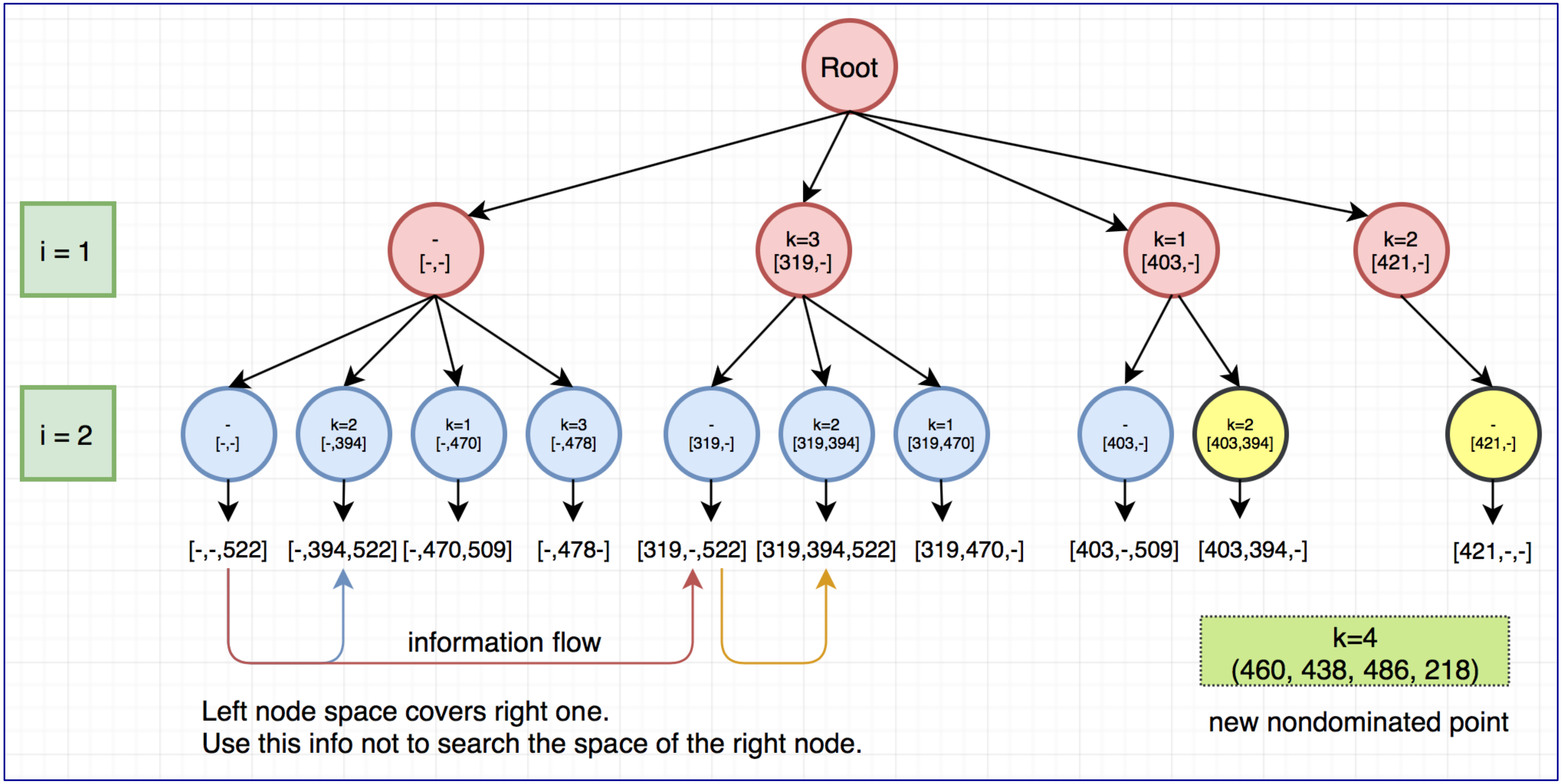
The Approach - Demo: Tree creation and update

Points	z_1	z_2	z_3	z_4
1	402	469	**521**	324
2	420	**393**	508	452
3	**318**	477	487	521



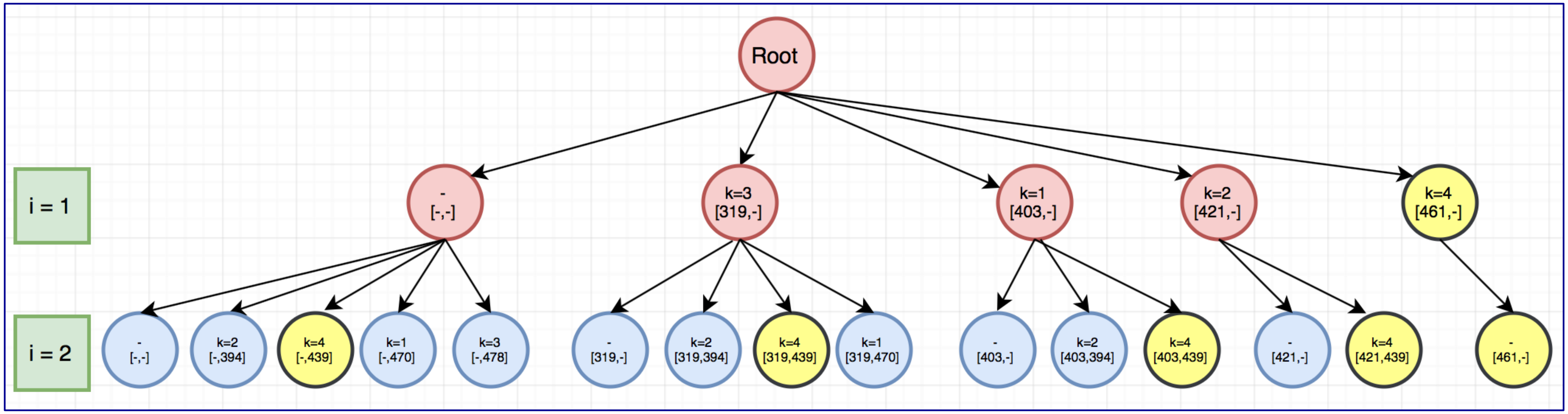
A web-based solution platform - nMOCO-S

The Approach - Demo: Search



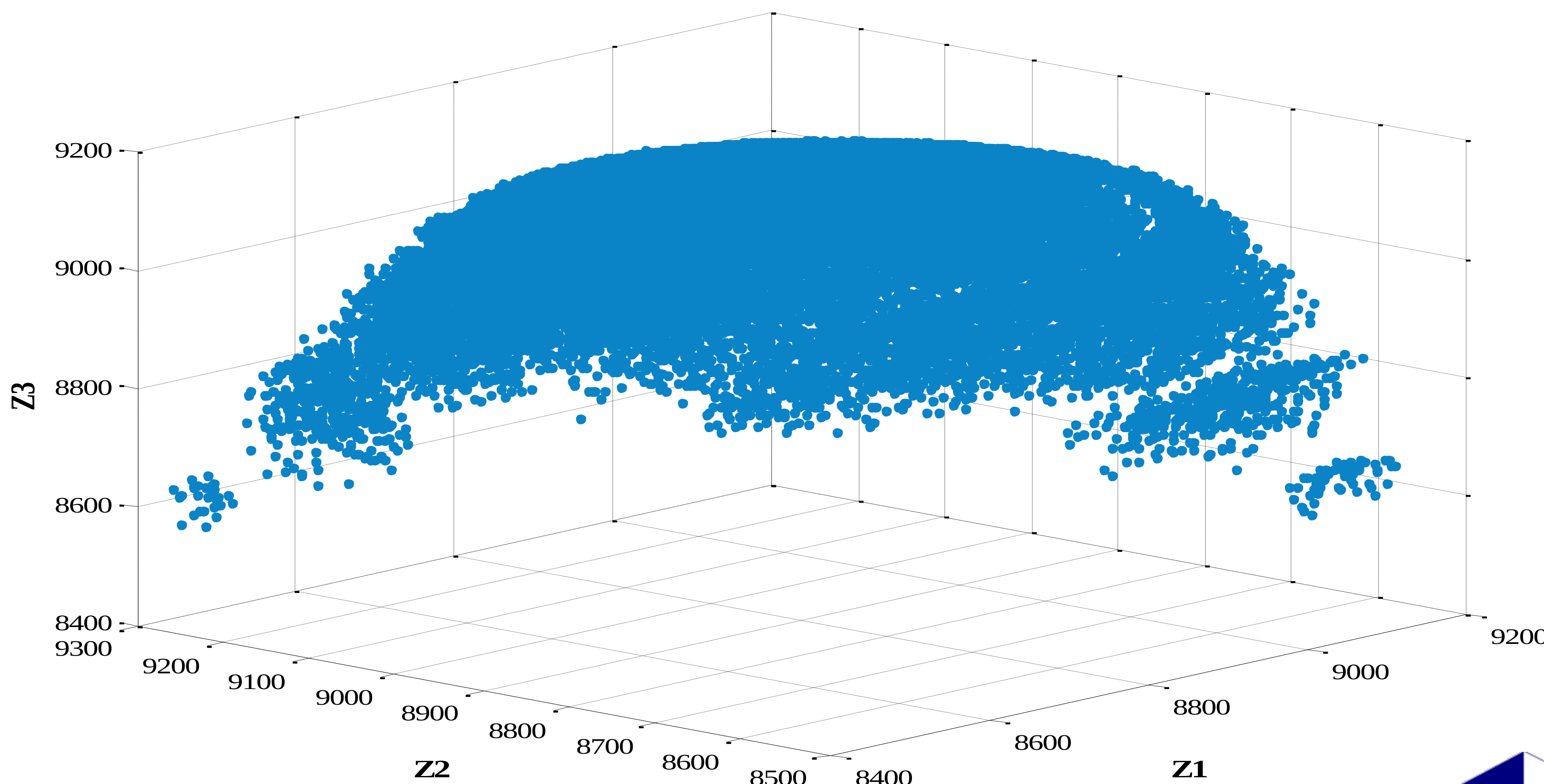
A web-based solution platform - nMOCO-S

The Approach - Demo: Tree update



A web-based solution platform - rMOCO-S

200-item MOKP with 27,260 nondominated points



A web-based solution platform - rMOCO-S

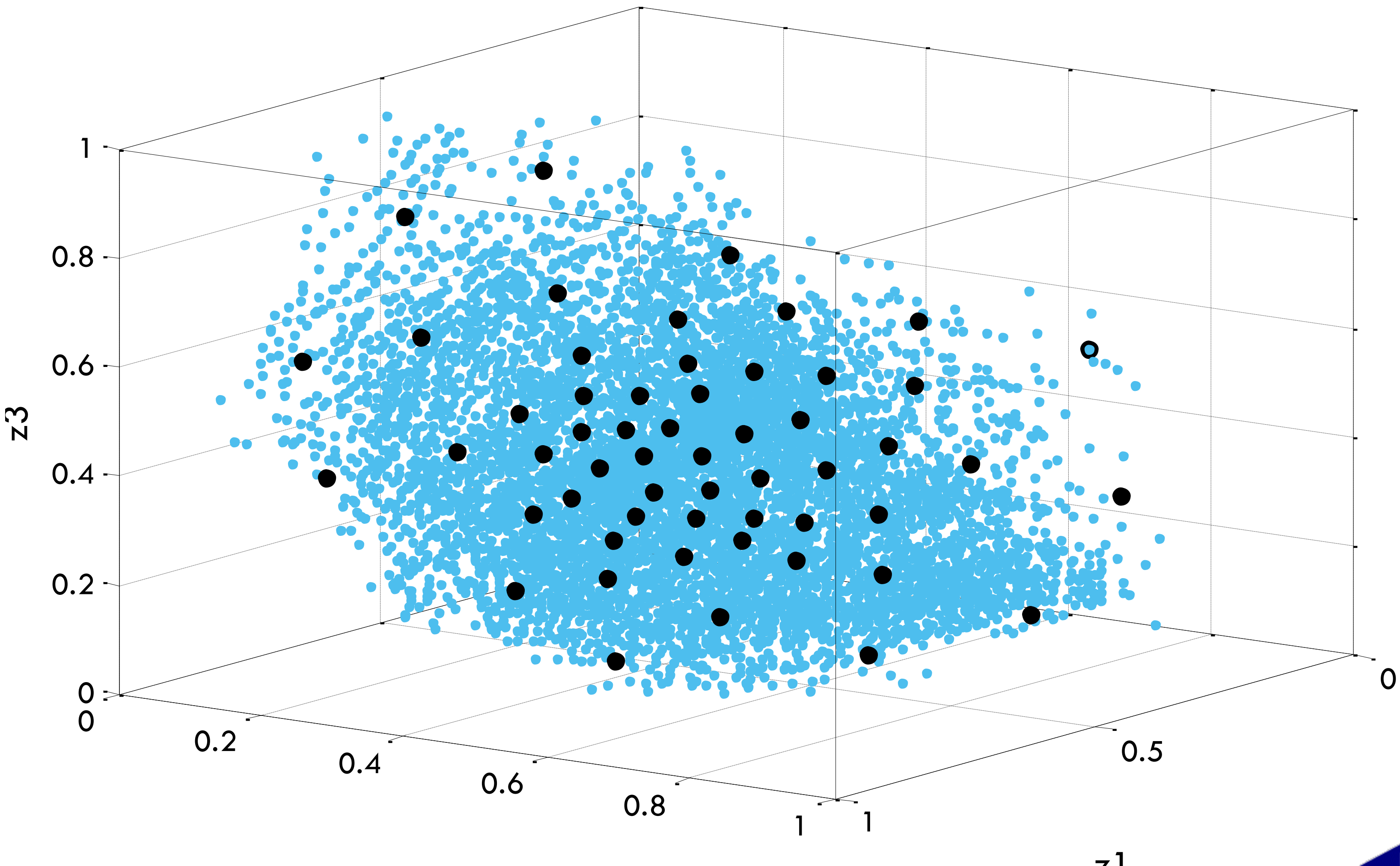
rMOCO-S

Representative nondominated points by Multi-objective Combinatorial Optimization Solver

Represent the objective space of your problem with a few points in an efficient manner.

Learn more.

A web-based solution platform - rMOCO-S

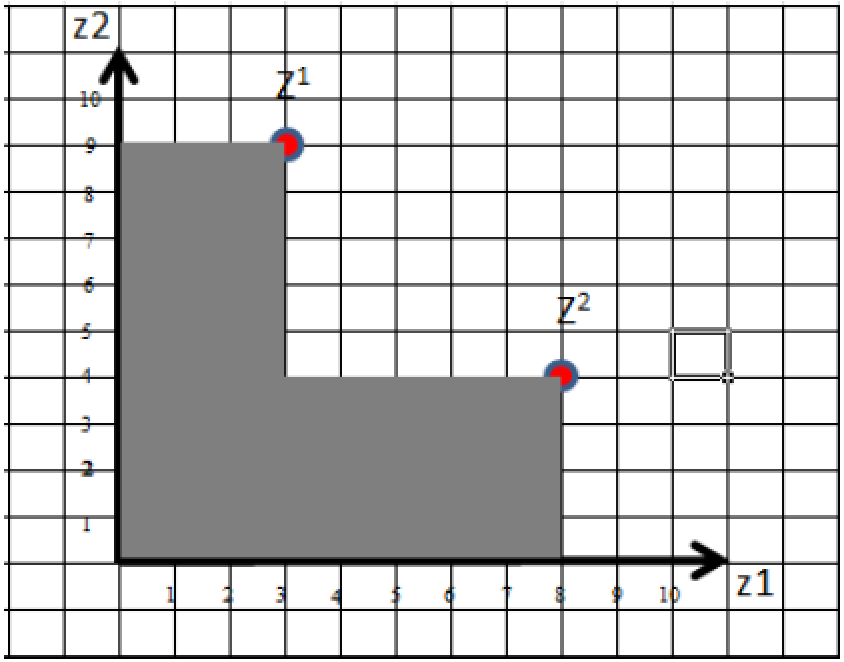


A web-based solution platform - rMOCOS

- Representative nondominated sets are found for MOIPs.
  - SBA is designed to continue generating new points until the desired coverage gap value is satisfied.
  - TDA guarantees to achieve a desired coverage gap value stated by the DM at the outset.
  - SPA is designed to consider the whole solution space and allocate the representative points accordingly if the number of representative points is known in advance.
- A desired level of quality is guaranteed in representing the nondominated frontier.
- The methods are computationally efficient based on extensive computational tests.
- **New approaches based on the distribution properties.**

A web-based solution platform - rMOCOS

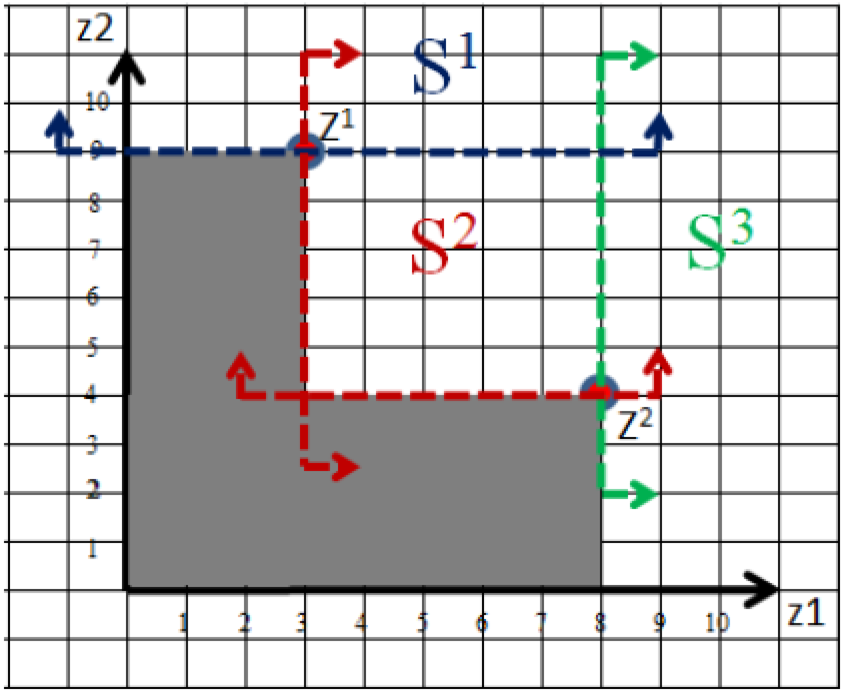
Sylva and Crema (2007)



$$
\begin{aligned}
\text{Max} \quad & \alpha + \epsilon(z_1(\mathrm{x}) + z_2(\mathrm{x})) \\
\text{s.to.} \quad & \\
& z_1(x) \geq z_1^1(\mathrm{x})y_1^1 + 1 + \alpha - M(1 - y_1^1) \\
& z_2(x) \geq z_2^1\mathrm{xx})y_2^1 + 1 + \alpha - M(1 - y_2^1) \\
& y_1^1 + y_2^1 = 1 \\
& z_1(x) \geq z_1^2(\mathrm{x})y_1^2 + 1 + \alpha - M(1 - y_1^2) \\
& z_2(x) \geq z_2^2(\mathrm{x})y_2^2 + 1 + \alpha - M(1 - y_2^2) \\
& y_1^2 + y_2^2 = 1 \\
& \mathrm{x} \in \mathbf{X} \\
& y_i^j \in \{0, 1\} \quad i = 1, 2, \quad j = 1, 2
\end{aligned}
$$

A web-based solution platform - rMOCOS

A Subspace-based Approach (SBA)



Decomposition to nondominated subspaces:

$$\mathbf{S}^j = (lb_1, lb_2)$$

$$\text{Max} \quad \alpha^j + \epsilon(z_1(\mathbf{x}) + z_2(\mathbf{x}))$$

s.to.

$$z_1(\mathbf{x}) \geq lb_1 + \alpha^j$$

$$z_2(\mathbf{x}) \geq lb_2 + \alpha^j$$

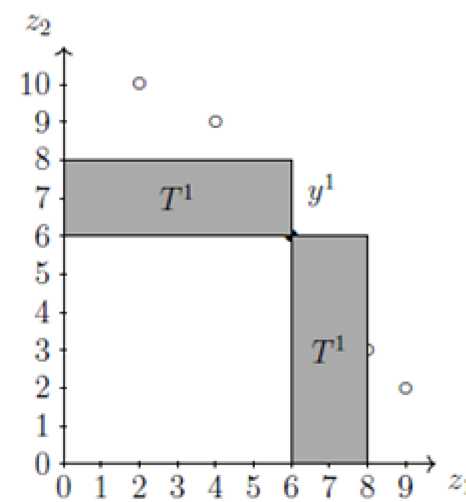
$$x \in X$$

$$\alpha^* = \max_{j=1,2,3} \{\alpha^j\}$$

A web-based solution platform - rMOCOS

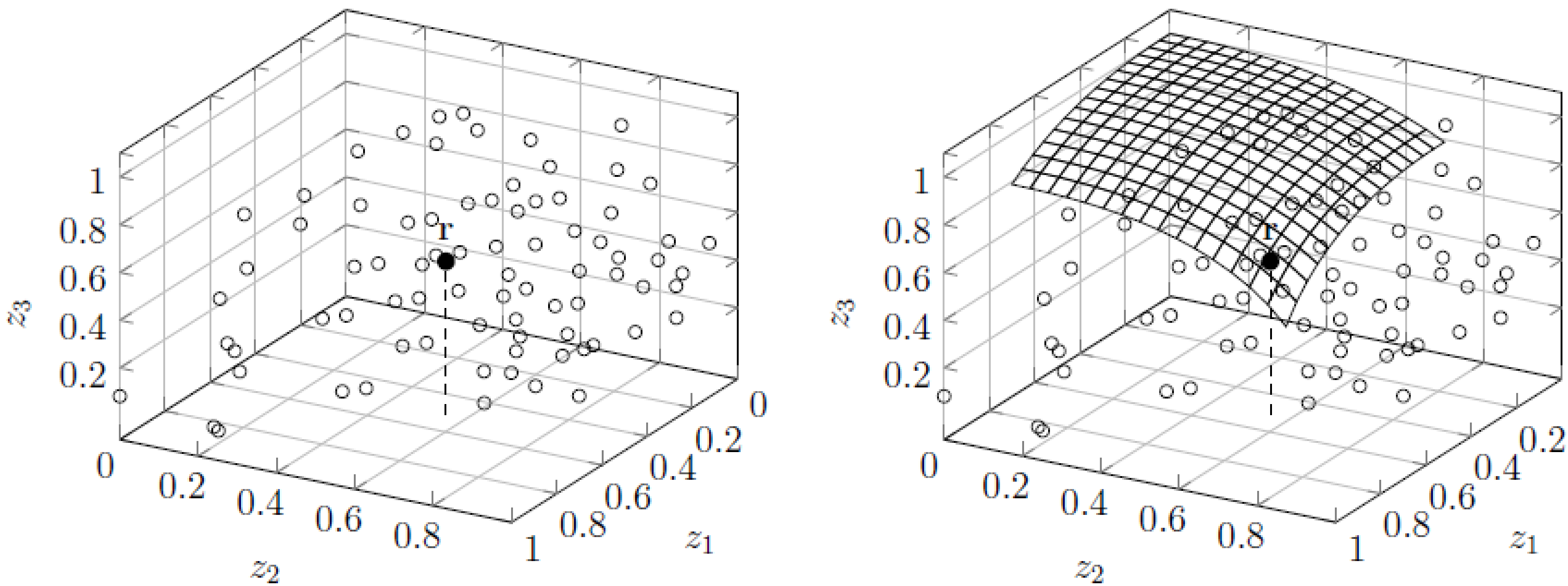
Territory Defining Algorithm (TDA)

- SBA reduces the computational effort compared to SC substantially.
- SBA controls the coverage gap value and stopped when the DM is satisfied.
- If a desired coverage gap value, Δ, is given at the outset, TDA
  - avoid generating points close to other points,
  - utilize the desired coverage gap information actively throughout the algorithm
  - constructs **territories** around the previously generated points that are inadmissible for the new point
  - keep searching different subspaces until finding a nondominated point.



A web-based solution platform - rMOCOS

Surface Projection Algorithm (SPA)



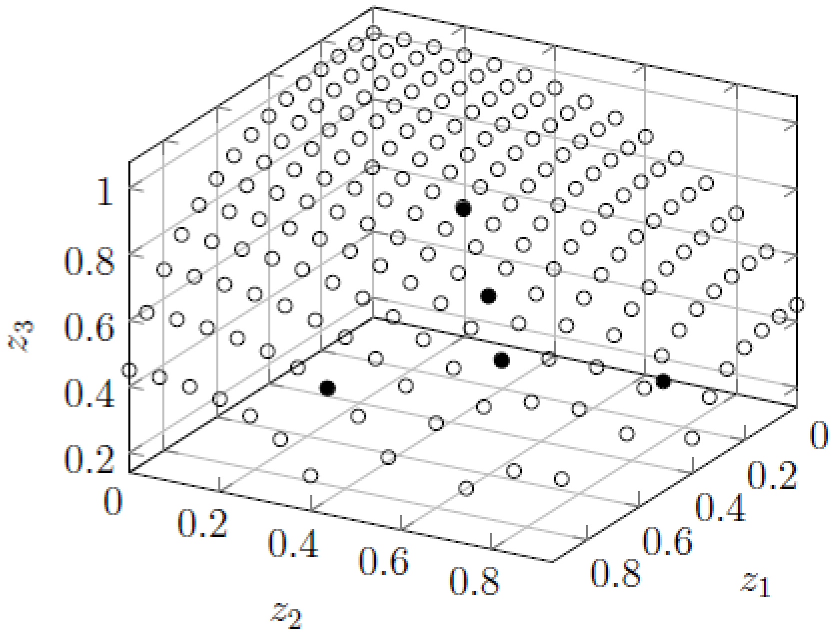
(a) All nondominated points

(b) A portion of the $L_{2.37}$ surface

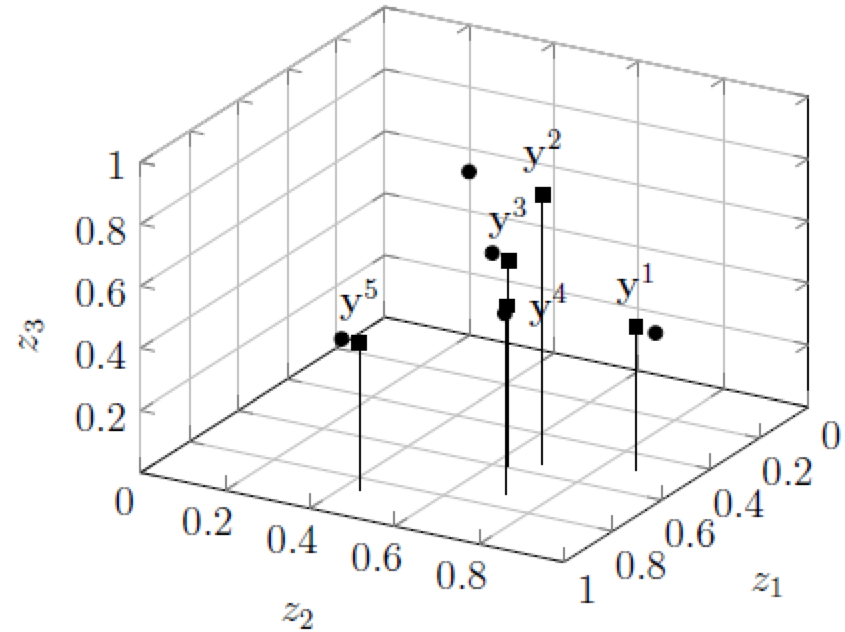
Approximate the nondominated frontier fitting Lp surface using the methodology of Koksalan and Lokman (2009).

A web-based solution platform - rMOCOS

Surface Projection Algorithm (SPA)



(c) Optimal subset of hypothetical points, $\alpha = 0.16$.

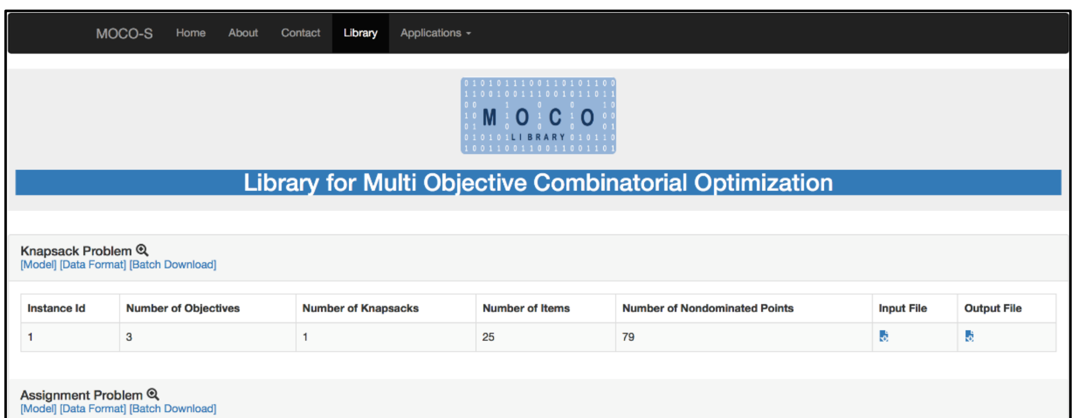


(d) The representative subset R, $\alpha_R = 0.18$.

Select a representative set of hypothetical points on the fitted surface.
(first discretize the surface and find the optimal subset.) For each hypothetical point, find the representative nondominated point at minimum Tchebycheff distance.

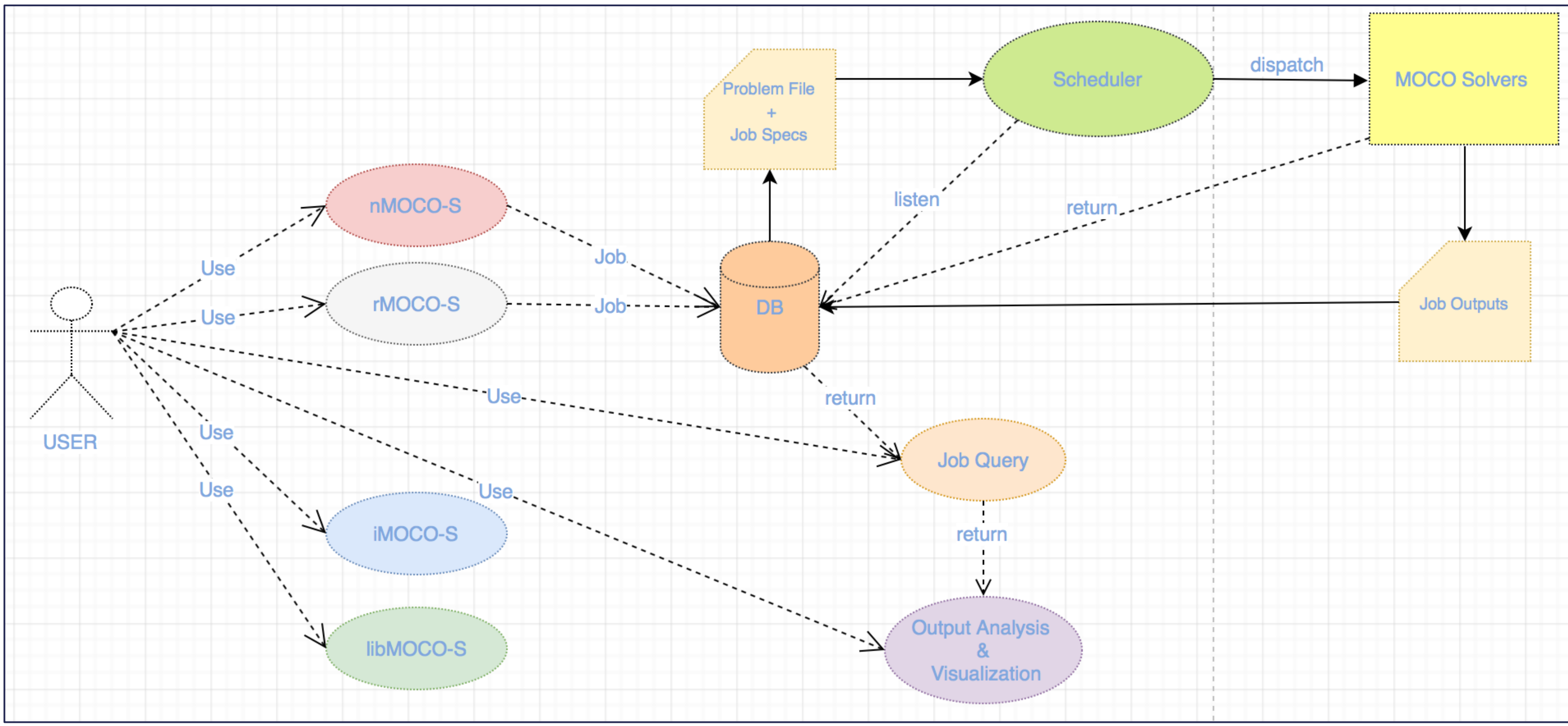
A web-based solution platform - Demo

- We implement these algorithms as **an online tool** that allows the users to generate:
  - a representative set satisfying a desired level of accuracy
  - all nondominated points.
- The tool provides the output in terms of the **objective function values** of the generated nondominated points as well as in terms of the **decision variables** corresponding to the desired nondominated points.
- We also maintain **a digital library** that contains a collection of MOIPs and make their inputs and outputs available to researchers



A web-based solution platform

Use Case:



A web-based solution platform - Demo

nMOCO-S: A MOKP with 3 objectives, 3 knapsacks, 25 items

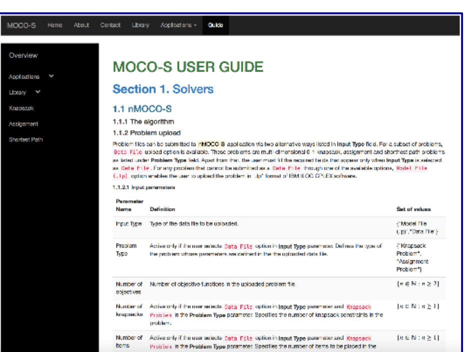
Step 1. Create a text file following the format given in the guide.

Items

obj. coeffs

weight coeffs and capacities

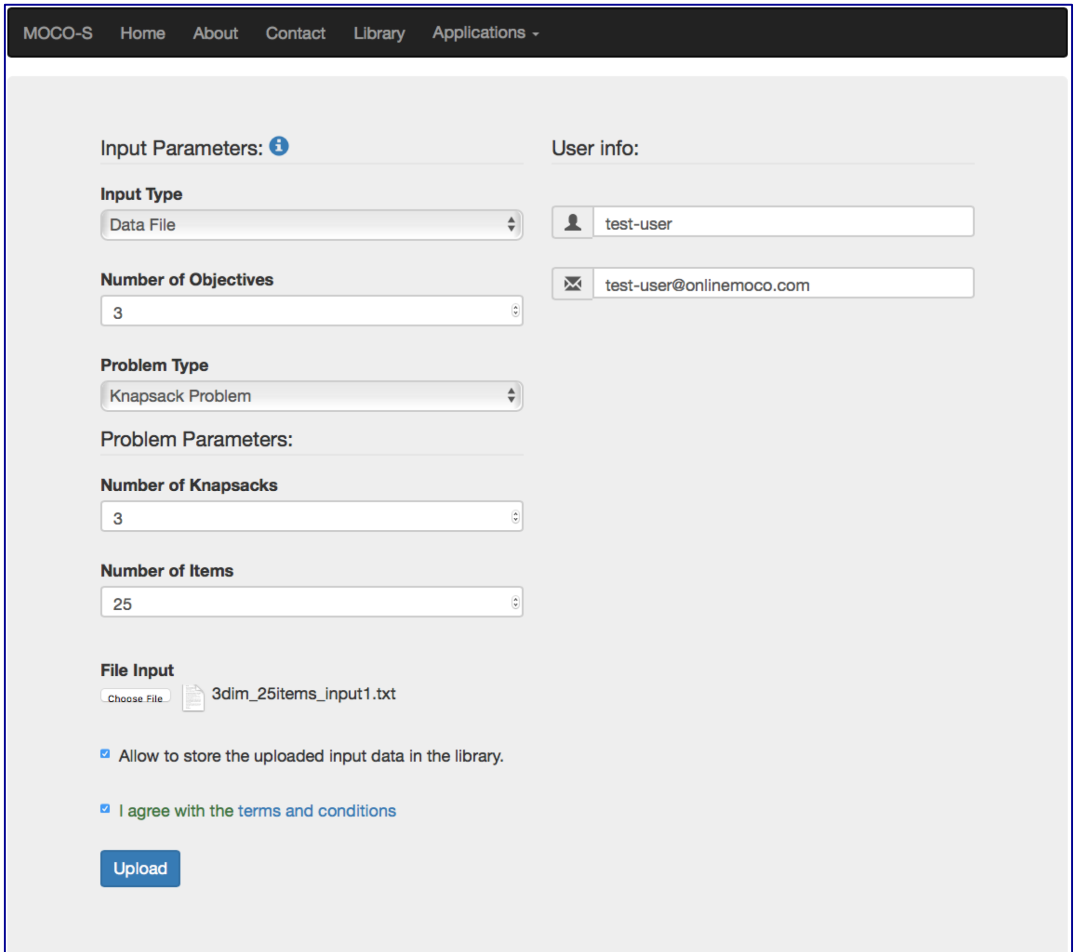
```
88 32 96 19 76 55 80 52 41 11 91 85 62 77 61 29 76 17 46 42 31 78 90 98 23
59 90 84 17 24 45 76 94 31 45 48 78 29 98 76 98 65 33 10 73 24 14 43 31 65
39 76 24 92 71 53 12 28 31 92 78 25 66 13 78 69 98 11 51 77 37 99 33 38 91
80 50 83 11 71 13 50 52 68 63 65 98 23 39 55 78 78 91 31 90 64 74 53 19 53
726
72 99 10 11 77 69 42 14 96 54 23 38 90 47 32 18 29 16 29 48 22 41 37 48 77
569
37 10 75 97 12 32 65 45 41 94 64 53 79 25 47 82 74 71 48 67 54 72 48 10 19
660
```



For any problem for which an input data format is not available, user can input ".lp" file of the single objective problem.

A web-based solution platform - Demo

Step 2. Upload the problem.



Applications→nMOCO-S

JobID →

Upload result:

Input data is valid!

Job Id: 22 (Keep this number to be able to query the status of your job!)

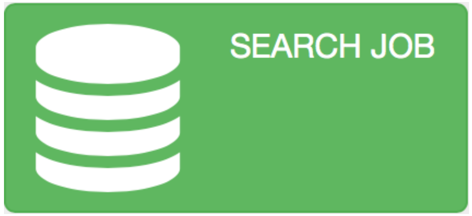
An email will be sent to the mail address **test-user@onlinemoco.com** as soon as the job is completed.

Return to nMOCO-S home page.

See the server queue and the status of your job.

A web-based solution platform - Demo

Step 3. Query the status of the problem with the user mail and given jobID.



MOCO-S Home About Contact Library Applications

test-user@onlinemoco.com 22 Search

Search Results

Job Id	Issuer	Job Creation Time	Job Status	Job Completion Time	Processing Time (secs)	Job Output
22	test-user@onlinemoco.com	2017-07-11 21:20:07.116	FINISHED_SUCCESS	2017-07-11 21:21:05.276	11.0	result.txt

# of nondominated points
# of models solved
elapsed time

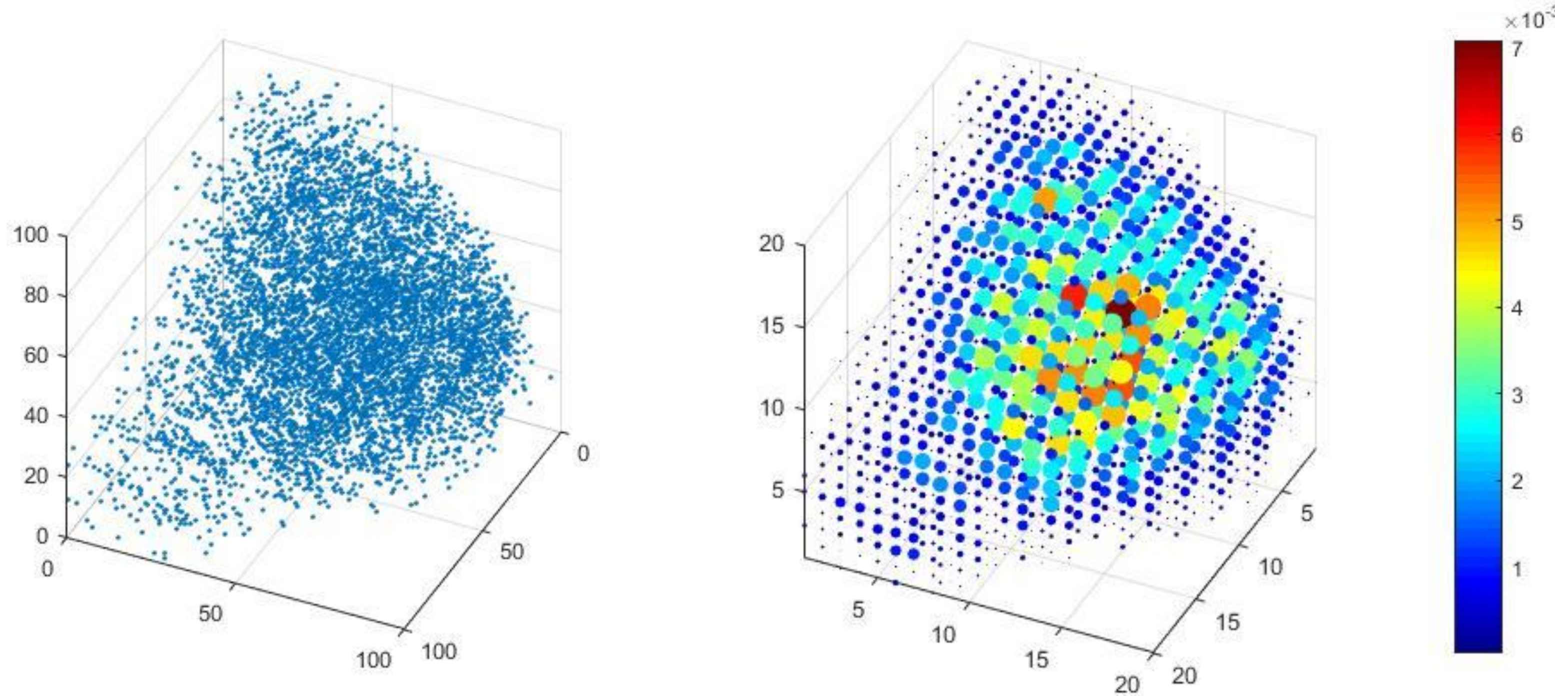
nondominated points

```
182
410
12.783
636 646 959
653 558 941
645 570 936
640 583 933
```

...

A web-based solution platform – Visualization Tools

Parzen Windows* (Ozarık, Koksalan and Lokman)

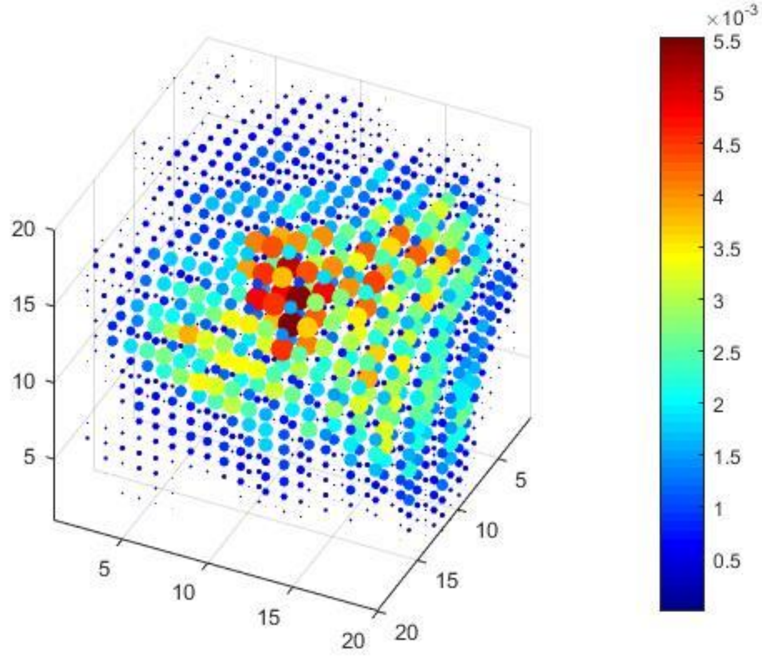


MOKP with 5652 nondominated points

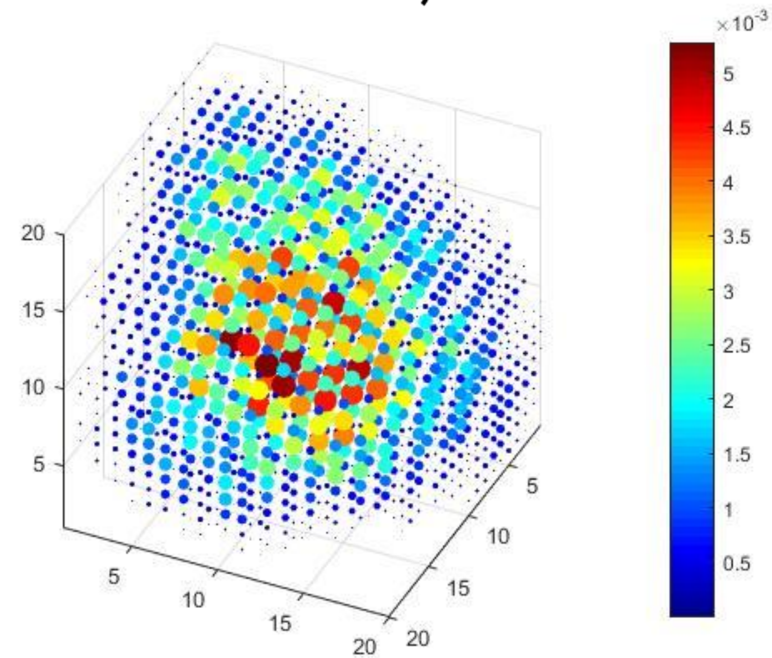
(See: Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley)

A web-based solution platform – Visualization Tools

Parzen Windows* (Ozarık, Koksalan and Lokman)



MOAP with 6573 nondominated points



MOKP with 6500 nondominated points

CONCLUSIONS

- A web-based decision support system for MOIPs generating:
  - Nondominated set (all or regional)
  - Representative sets

  based on a variety of algorithms.
- Available to academic researchers.
- A digital library.

FUTURE WORK

- New algorithms that consider
  - The distribution and density
  - Shape of the frontier
  - Preferences of the DM.
  - New quality measures.

Thank you...

Questions & Comments

References

Boland, N., Charkhgard, H., Savelsbergh, M. 2015. “A Criterion space search algorithm for biobjective integer programming: The Balanced Box Method”, INFORMS Journal on Computing, vol. 27 (4), 735-754.

Boland, N., Charkhgard, H., Savelsbergh, M. 2016. “The L-shape search method for triobjective integer Programming,” Mathematical Programming Computation, vol. 8 (2), 217-251.

Ceyhan, G. 2014. “Generating representative nondominated point subsets in multiobjective integer programs,” M.S. Thesis, Middle East Technical University, Ankara.

Dächert, K., Klamroth, K. 2015. “A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems,” Journal of Global Optimization, vol. 61(4), 643-676.

Dhaenens, C., Lemesre, J., Talbi, E.G., 2010, “K-PPM: A new exact method to solve multi-objective combinatorial optimization problems,” European Journal of Operational Research, vol. 200, 45-53.

Ehrgott, M., Gandibleux, X. 2000. “A survey and annotated bibliography of multiobjective combinatorial optimization,” OR Spectrum, vol. 22(4), 425-460.

Ehrgott, M., Gandibleux, X. 2004. “Approximative solution methods for multiobjective combinatorial optimization,” Sociedad de Estadística E Investigación Operativa, vol. 12(1), 1-89.

Ehrgott, M., X. Gandibleux. 2008. “Hybrid metaheuristics for multi-objective combinatorial optimization,” Studies in Computational Intelligence (SCI), vol. 114, 221-259.

Sylva, J., Crema, A. 2004. “A method for finding the set of nondominated vectors for multiple objective integer linear programs”, European Journal of Operational Research, vol. 158, 46-55.

Sylva, J., Crema, A. 2007. “A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs,” European Journal of Operational Research, vol. 180(3), p. 1011-1027.

Mavrotas, G., Florios, K. 2013. “An improved version of the augmented e-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems”, Applied Mathematics and Computation, vol. 219, 9652-9669.

Özlen, M., Azizoglu, M., 2009. “Multi-objective integer programming: a general approach for generating all non-dominated solutions,” European Journal of Operational Research., vol. 199, 25-35.

Özlen, M., Burton, B. A., MacRae, C. A. G., 2014. “Multi-objective integer programming: an improved recursive algorithm,” Journal of Optimization Theory and Applications, vol. 160 (2), 470-482.

Przybylski, A., Gandibleux, X., Ehrgott, M., 2010. “A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives, ” Discrete Optimatization, vol. 7, 149-165.

.

EXACT ALGORITHMS (Lokman&Köksalan, JOGO'13)

Comparison of Algorithms on MOKP for q = 3

Number of items	Problem	Number of nondominated points (N)	Solution time (CPU time in seconds)		
			Sylva and Crema	Algorithm 1	Algorithm 2
20	1	35	14.24	6.75	3.62
	2	43	38.47	15.18	5.41
	3	61	102.40	39.29	8.80
	4	67	121.82	31.34	8.31
	5	77	259.51	48.76	10.59
25	1	57	118.13	40.73	9.65
	2	76	314.61	54.80	10.70
	3	103	818.26	53.35	15.08
	4	108	2,043.33	192.63	25.15
	5	132	5,291.38	193.52	20.67
	6	157	5,285.43	276.52	33.27
	7	163	5,253.49	245.59	25.56
	8	168	12,406.04	551.48	38.16
	9	182	14,740.24	407.65	30.23
	10	470	Could not be solved in 15 h	1,619.32	44.75

EXACT ALGORITHMS (Lokman&Köksalan, JOGO'13)

Comparison of Algorithms on MOKP for q = 3

Number of items	Problem	Number of nondominated points (N)	Solution time (CPU time in seconds)	
			Algorithm 1	Algorithm 2
50	1	280	4,823.95	121.82
	2	356	5,173.84	139.63
	3	519	12,082.91	186.40
	4	784	33,699.41	360.54
	5	912	35,557.58	383.64

EXACT ALGORITHMS (Lokman&Köksalan, JOGO'13)

Performance of Algorithm 2 (q=3)

- For q=3, the number of models solved, MS, to find all N nondominated points will be in the interval:

$$N + 1 \leq MS \leq (N + 1)(N + 2)/2.$$

- We observe that MS/N is in the interval [1.80, 2.36] with an average of 2.13.
- That is, we roughly solve **only 2 models for each nondominated point** on average.
- This indicates the importance of the information obtained from the archives of Algorithm 2.
- MS << (N + 1)(N + 2)/2 especially for large N values.

EXACT ALGORITHMS (Lokman&Köksalan, JOGO'13)

Performance of Algorithm 2 (q=4)

- MS/N is again not sensitive to the value of N where the ratio is within the interval [5.67,10.14] with an average value of 8.53.
- The value of MS/N increases with the number of objectives.

$$N \le MS \le \sum_{n=0}^{N} \frac{(n+1)(n+2)}{2}$$

- In our experiments: $MS \cong 9N \quad (MS << \sum_{n=0}^{N} \frac{(n+1)(n+2)}{2})$

EXACT ALGORITHMS (Lokman&Köksalan, JOGO'13)

Performance of Algorithm 2

- Still may not be practical for large problems.
- Number of nondominated points may be prohibitive.
- Can be used to test performances of heuristics.

e.g. 51.28 hours to solve a MOKP with 200 items and 3 objectives (27,260 nondominated points)

A web-based solution platform - rMOCOS

The algorithm of Sylva and Crema (SC)

$$
\begin{aligned}
&\text{Max} \quad F = \alpha + \epsilon \sum_{i=1}^{m} w_i z_i \\
&\text{s.to.} \\
&\qquad z_i \geq y_i^j \gamma_{ji} + \alpha - (M_i + U)(1 - \gamma_{ji}) \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n \\
&\qquad \sum_{i=1}^{m} \gamma_{ji} = 1 \qquad j = 1, \ldots, n \\
&\qquad \alpha \geq 0 \\
&\qquad \gamma_{ji} \in \{0, 1\} \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n \\
&\qquad \mathrm{z} = \mathrm{z}(\mathrm{x}) \\
&\qquad \mathrm{x} \in X
\end{aligned}
$$

A Subspace-based Approach (SBA)

Solution time comparison of SBA and SC
on 3-objective, 50-item knapsack problems (in secs)*

	SBA		SC	
$\|R\|$	Avg.	StDev.	Avg.	StDev.
5	1.40	1.01	0.47	0.03
50	36.83	10.51	81.43	27.10
100	86.37	34.77	3863.25	1989.54
120	$\boxed{103.54}$	44.75	$\boxed{15291.65}$	8096.38

**Based on 10 problems per cell and there are an average of 417 nondominated points in total.*

A Subspace-based Approach (SBA)

Problem	$\|R\|$	Coverage gap Avg.	Coverage gap StDev.	Sol. time (sec) Avg.	Sol. time (sec) StDev.	Models solved Avg.	Models solved StDev.
KP-25	5	0.20	0.04	0.64	0.17	16.60	1.84
	25	0.05	0.02	7.4	1.78	164.88	29.51
$(\bar{N} = 56)$	50	0.02	0.01	15.87	1.31	356.50	32.50
KP-50	5	0.22	0.03	1.4	1.01	18.40	1.07
	25	0.08	0.02	12.74	2.56	224.10	34.00
	50	0.04	0.01	36.83	10.51	639.80	161.49
$(\bar{N} = 417)$	100	0.02	0.01	86.37	34.77	1471.90	518.16
KP-100	5	0.24	0.02	1.24	0.51	18.60	1.43
	25	0.10	0.01	20.53	2.31	280.60	21.26
	50	0.05	0.00	76.75	9.5	1013.20	97.46
$(\bar{N} = 3289)$	100	0.00	0.00	$\boxed{227.33}$	29.56	2889.60	239.35

A Subspace-based Approach (SBA)

Problem	$\lvert R \rvert$	Coverage gap Avg.	Coverage gap StDev.	Sol. time (sec) Avg.	Sol. time (sec) StDev.	Models solved Avg.	Models solved StDev.
AP-10 ($\bar{N} = 185$)	5	0.22	0.05	0.93	0.46	17.50	2.37
	25	0.07	0.02	11.23	2.44	212.10	31.66
	50	0.04	0.02	24.88	7.15	446.80	117.07
	100	0.02	0.01	48.81	18.21	857.25	274.26
AP-20 ($\bar{N} = 1548$)	5	0.23	0.03	1.61	0.55	18.40	1.71
	25	0.10	0.02	24.96	4.24	260.60	38.65
	50	0.06	0.01	82.88	15.92	829.70	139.71
	100	0.04	0.01	219.94	46.59	2116.50	386.92
AP-30 ($\bar{N} = 5372$)	5	0.27	0.04	2.13	0.37	17.90	1.79
	25	0.12	0.02	46.04	7.74	300.00	34.65
	50	0.07	0.01	169.99	38.71	1034.10	164.17
	100	0.04	0.01	546.14	140.82	3175.60	544.03

Territory Defining Algorithm (TDA)

Problem	Threshold	$\lvert R \rvert$ Avg.	$\lvert R \rvert$ StDev.	Models solved (%) Avg.	Models solved (%) StDev.	Solution time (%) Avg.	Solution time (%) StDev.
KP-25	Δ_5	6.20	1.32	84.66	16.62	117.80	70.55
	Δ_{25}	24.75	1.83	45.09	9.03	50.29	13.85
	Δ_{50}	47.83	3.43	43.22	5.55	55.85	7.79
KP-50	Δ_5	6.60	1.51	74.36	12.16	60.09	39.15
	Δ_{25}	22.60	4.55	28.63	9.06	23.55	7.51
	Δ_{50}	47.70	3.65	23.55	5.50	21.36	7.03
	Δ_{100}	90.40	5.87	23.90	8.98	25.34	13.17
KP-100	Δ_5	7.50	0.97	81.21	13.15	79.53	28.58
	Δ_{25}	20.40	4.01	18.72	2.76	12.66	3.75
	Δ_{50}	53.00	11.22	15.36	2.17	10.78	2.22
	$\boxed{\Delta_{100}}$	97.80	10.12	10.57	1.23	$\boxed{8.13}$	1.47

Models solved and solution time values show the number of models solved and solution times of TDA as percentages of those values of SBA, respectively

Territory Defining Algorithm (TDA)

		$\lvert R \rvert$		Models solved (%)		Solution time (%)	
Problem	Threshold	Avg.	StDev.	Avg.	StDev.	Avg.	StDev.
AP-10	Δ_5	6.20	1.23	76.98	14.49	65.03	36.29
	Δ_{25}	22.50	2.88	31.99	7.79	24.97	7.74
	Δ_{50}	41.30	3.74	31.05	7.76	30.27	19.02
	Δ_{100}	83.88	4.85	34.04	9.46	32.30	10.55
AP-20	Δ_5	7.20	1.40	82.73	26.38	61.37	23.28
	Δ_{25}	18.50	3.47	20.04	8.39	14.67	6.30
	Δ_{50}	41.90	5.11	16.76	3.43	12.51	2.96
	Δ_{100}	88.50	18.90	14.82	4.11	12.11	3.78
AP-30	Δ_5	7.00	1.41	74.43	18.74	60.86	22.13
	Δ_{25}	18.10	4.18	15.43	3.08	10.74	2.53
	Δ_{50}	41.00	4.92	12.07	2.95	8.78	2.29
	$\boxed{\Delta_{100}}$	87.60	11.05	9.17	2.50	$\boxed{6.95}$	2.09

Models solved and solution time values show the number of models solved and solution times of TDA as percentages of those values of SBA, respectively

A web-based solution platform - nMOCO-S

The Approach - Tree node data

- Node: **k**
- Depth of node: **i**
- Branching point: $\mathbf{z}^k$
- Branching criterion: $\mathbf{z}^k_i$
- Parent node: **m** with bound vector $\mathbf{b}^m$
- Bounds: $(\mathbf{b}^m_1 , \mathbf{b}^m_2 ,\ldots, \mathbf{b}^m_{i-1} , \mathbf{b}^k_i , - , \ldots)$
- All siblings at depth *i* have first (i-1) bounds in common. Right sibling has a tighter bound than the one at left at criterion *i*.
- No bound set for criteria after index *i*.
- Leaf nodes have bounds at each criterion except the last one and completely defines the search region.

A web-based solution platform - nMOCO-S

The Approach - Tree node elements

- **Node:** defines the search space for an additional nondominated point
- **Branching point:** which nondominated point to be used to define a bound for the search space.
- **Branching criterion:** the criterion whose index is equal to the depth of the node
- **Parent node:** search space that contains the search space of the current node and its siblings
- **Bounds:** set by the *branching criterion* values of *branching points* of the current node and its parents

A web-based solution platform

www.onlinemoco.com/MOIP/

- In development phase.
- Applications:
  - **nMOCO-S:** Finds all nondominated points.
  - **rMOCO-S:** Finds a representative set of nondominated points
  - **iMOCO-S:** Integrates user preferences
  - **libMOCO-S:** Collects a variety of MOCO instances.
- Client side technologies: **html5, javascript, jQuery**
- Server side technologies: **java servlet**
- Web server: **Apache Tomcat**
- Database: **Apache Derby** and **JDBC**